



TUGAS AKHIR - TF 141581
PERANCANGAN SISTEM KONTROL SUDUT
***PITCH* TURBIN ANGIN MENGGUNAKAN**
METODE PSO
(PARTICLE SWARM OPTIMIZATION)

JAUHAROTUL MAKNUNAH
NRP.2414106028

Dosen Pembimbing
Dr. Ir. Ali Musyafa', MSc.
Dr. Imam Abadi, ST. MT.

DEPARTEMEN TEKNIK FISIKA
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2017



FINAL PROJECT - TF141581
DESIGN OF PITCH ANGLE CONTROL SYSTEM
ON WIND TURBINE USING PSO
(PARTICLE SWARM OPTIMIZATION) METHOD

JAUHAROTUL MAKNUNAH
NRP.2414106028

Supervisors
Dr. Ir. Ali Musyafa', MSc.
Dr. Imam Abadi, ST. MT.

DEPARTMENT OF ENGINEERING PHYSICS
Faculty of Industrial Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2017

**PERANCANGAN SISTEM KONTROL SUDUT *PITCH*
TURBIN ANGIN MENGGUNAKAN METODE PSO
(*PARTICLE SWARM OPTIMIZATION*)**

TUGAS AKHIR

Oleh:

Jauharotul Maknunah

NRP : 2414 106 028

Surabaya, 25 Januari 2017

Mengetahui / Menyetujui

Pembimbing



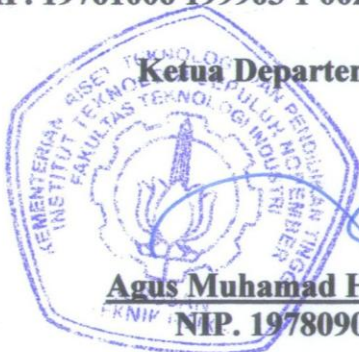
Dr. Imam Abadi, S.T., M.T.
NIP. 19761006 199903 1 002

Pembimbing



Dr. Ir. Ali Musyafa', MSc.
NIP. 19600901 198701 1 001

Ketua Departemen Teknik Fisika



Agus Muhamad Hatta, ST, MSi, Ph.D
NIP. 19780902 200312 1 002

**PERANCANGAN SISTEM KONTROL SUDUT *PITCH*
TURBIN ANGIN MENGGUNAKAN METODE PSO
(*PARTICLE SWARM OPTIMIZATION*)**

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Teknik
pada
Bidang Studi Rekayasa Instrumentasi
Program Studi S-1 Departemen Teknik Fisika
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember

Oleh :

Jauharotul Maknunah
NRP. 2414106028

Disetujui oleh Tim Penguji Tugas Akhir :

1. Dr. Ir. Ali Musyafa', MSc.(Pembimbing 1)
2. Dr. Imam Abadi, S.T., M.T.(Pembimbing 2)
3. Dr.Ir. Purwadi Agus Darwito, M.Sc.(Penguji 1)
4. Prof. Dr. Ir. Aulia Siti Aisjah, M.T.(Penguji 2)
5. Ir. Wiratno Argo Asmoro, M.Sc.(Penguji 3)

SURABAYA
Januari, 2017

PERANCANGAN SISTEM KONTROL SUDUT *PITCH* TURBIN ANGIN MENGGUNAKAN METODE PSO (*PARTICLE SWARM OPTIMIZATION*)

Nama Mahasiswa : Jauharotul Maknunah
NRP : 2414106028
Program Studi : S1 Teknik Fisika
Dosen Pembimbing : Dr. Ir. Ali Musyafa', MSc.
Dr. Imam Abadi, ST. MT.

Abstrak

Blade pada turbin angin bergerak dikarenakan adanya gaya *lift* dan gaya *drag* yang dihasilkan. Selama proses konversi energi gerak angin menjadi energi listrik, tidak dipungkiri terdapat *loss energy* atau energi yang hilang. Salah satu teknik optimalisasi daya pada turbin angin yakni dengan menerapkan penggunaan kontrol sudut *pitch* pada bilah turbin agar keluaran daya turbin angin tetap pada kondisi optimal. Pada perancangan turbin angin tersebut, komponen yang dibutuhkan adalah *Blade* turbin angin, motor servo sebagai aktuator, rotational connector serta *rotary encoder* sebagai *transmitter*. Hasil pengujian *prototype* turbin angin menunjukkan nilai *Power Coefficient* (CP) terbesar 0,57510. Dengan menerapkan metode kontrol PID (Proporsional Integral Derivatif), didapatkan nilai Kp sebesar 0,0239069, nilai Ki sebesar 0,0001382 serta nilai Kd sebesar 0,053518. Nilai parameter tersebut didapatkan dengan menggunakan metode optimalisasi PSO dengan jumlah iterasi sebanyak 50 Iterasi. Dari hasil uji dengan set point 40, 60, 80 dan 100 rpm, didapatkan bahwa sistem mampu mencapai *set point* dengan *overshoot* maksimum mencapai 26,25% dan nilai *rise time* lebih cepat jika dibandingkan tanpa controller yakni 181 detik.

Kata kunci : Turbin Angin, CP, PSO, PID

DESIGN OF PITCH ANGLE CONTROL SYSTEM ON WIND TURBINE USING PSO (PARTICLE SWARM OPTIMIZATION) METHOD

***Name : Jauharotul Maknunah
NRP : 2414106028
Study Program : S1 Teknik Fisika
Supervisor : Dr. Ir. Ali Musyafa ', MSc.
Dr. Imam Abadi, ST. MT.***

Abstract

Wind turbine blade on a move because of the lift force and drag force is generated. During the process of conversion on wind energy motion into electrical energy, there is no doubt that there is a loss of energy. One power optimization techniques in wind turbines is by implementing the use of controls on the turbine blade pitch angle so that the power output of the wind turbines in optimal conditions. In the design of the wind turbine, the component that needed is a wind turbine blade, servo motors as actuators, rotational connector and a rotary encoder as transmitter. The test results, the maximum value of wind turbine prototype Power Coefficient (CP), 0.57510. By applying the control method PID (Proportional Integral Derivative), Kp values obtained by 0.0239069, 0.0001382 amounting Ki value and Kd value of 0.053518. The parameter values obtained using PSO optimization method with the number of iterations is 50. From the set point test of 40, 60, 80 and 100 rpm, it was found that the system is able to achieve the set point with a maximum overshoot reached 26.25% with rise time faster than without the controller that is 181 seconds.

Keywords : Wind Turbine, CP, PSO, PID

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah S.W.T, karena rahmat dan hikmat-Nya sehingga penulis diberikan kesehatan, kemudahan, dan kelancaran dalam menyusun laporan tugas akhir ini. Tidak lupa juga penulis menyampaikan ucapan terima kasih kepada keluarga dan para sahabat. Oleh karena dukungan mereka, penulis mampu menyusun laporan tugas akhir yang berjudul:

“PERANCANGAN SISTEM KONTROL SUDUT PITCH TURBIN ANGIN MENGGUNAKAN METODE PSO (*PARTICLE SWARM OPTIMIZATION*)”

Tugas akhir ini merupakan salah satu persyaratan akademik yang harus dipenuhi dalam Program Studi S-1 Teknik Fisika FTI-ITS. Penulis menyampaikan terima kasih yang sebesar-besarnya kepada:

1. Agus Muhamad Hatta, ST, MSi, Ph.D. selaku ketua departemen Teknik Fisika ITS.
2. Segenap keluarga yang telah memberikan dukungan penuh dalam penyelesaian tugas akhir ini.
3. Dr. Ir. Ali Musyafa', MSc. selaku dosen pembimbing pertama tugas akhir ini, yang selalu memberikan semangat dan ide-ide baru.
4. Dr. Imam Abadi, S.T, M.T., MSc. selaku dosen pembimbing kedua tugas akhir ini, yang selalu memberikan semangat dan ide-ide baru.
5. Segenap Bapak/Ibu dosen pengajar di departemen Teknik Fisika - ITS.
6. Rekan-rekan LJ GENAP TF 2014 dan warga Teknik Fisika - ITS, yang senantiasa memberikan motivasi dan perhatian.
7. Teman-teman seperjuangan TA yang telah memotivasi dan memberikan bantuan-bantuan dalam penyelesaian laporan tugas akhir ini.
8. Rekan-rekan dan laboran dari Laboratorium Rekayasa Instrumentasi dan Laboratorium Pengukuran Fisis Teknik Fisika - ITS.

9. Teman-teman Paduan Suara Mahasiswa ITS dan Paduan Suara Mahasiswa PENS yang memberikan semangat lebih dalam menuntut ilmu.
10. Rekan-rekan kerja di BajuSenamGrosir.com yang senantiasa membantu dalam kelancaran menyelesaikan tugas akhir.
11. Teman-teman organisasi di luar kampus yang membuat saya semangat untuk terus menyelesaikan tugas akhir ini.

Penulis menyadari bahwa mungkin masih ada kekurangan dalam laporan ini, sehingga kritik dan saran penulis terima. Semoga laporan ini dapat berguna dan bermanfaat bagi penulis dan pihak yang membacanya.

Surabaya, 7 Januari 2017

Penulis

DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBAR PENGESAHAN.....	iii
ABSTRAK.....	v
<i>ABSTRACT</i>	vi
KATA PENGANTAR.....	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xiii
 BAB I PENDAHULUAN.....	 1
1.1. Latar Belakang.....	1
1.2. Perumusan Masalah.....	2
1.3. Tujuan Tugas Akhir.....	2
1.4. Batasan Masalah.....	3
 BAB II TINJAUAN PUSTAKA.....	 5
2.1. Turbin Angin.....	5
2.2. Komponen Turbin Angin.....	9
2.3. Kontroler PID (<i>Proportional Intregal Derivative</i>)....	15
2.4. PSO (<i>Particle Swarm Optimzation</i>).....	16
 BAB III METODOLOGI PENELITIAN.....	 19
3.1. Prosedur Umum Penelitian.....	19
3.2. Perancangan Turbin Angin.....	26
3.3. Perancangan Kontrol <i>Pitch</i> Turbin Angin.....	27
3.4. Pencarian Parameter Kontroler PID menggunakan PSO.....	29
3.5. Realisasi Kontrol Sudut <i>Pitch</i> Turbin Angin.....	30
 BAB IV HASIL DAN PEMBAHASAN.....	 33
4.1 Hasil Pengujian Perancangan <i>Hardware</i>	33
4.2 Perancangan Sistem Kontrol <i>Pitch</i> Turbin Angin.....	33
4.3 Tuning PID Menggunakan Metode PSO.....	39

4.4 Pengujian Tracking <i>Set Point</i> Sistem Kontrol Turbin Angin.....	44
BAB V PENUTUP.....	53
5.1 Kesimpulan.....	53
5.2 Saran.....	53
DAFTAR PUSTAKA	
LAMPIRAN A Program <i>Tunnig</i> PID – PSO	
LAMPIRAN B Program Arduino	
LAMPIRAN C Respon <i>Real Plant</i> Turbin Angin dengan <i>Input</i> Sudut Motor Servo 4 ⁰ dan Kecepatan Angin 2,6 m/s	
BIODATA PENULIS	

DAFTAR GAMBAR

Gambar 2.1	<i>Tube</i> angin yang melewati turbin angin.....	6
Gambar 2.2	Fenomena <i>Drag</i> dan <i>Lift</i>	7
Gambar 2.3	Bentuk fisik motor servo.....	10
Gambar 2.4	Mekanisme <i>Slip Ring</i>	11
Gambar 2.5	Desain Umum <i>Rotary Encoder</i>	12
Gambar 2.6	Papan Arduino	13
Gambar 2.7	Blok Diagram Pengendali PID	15
Gambar 3.1	<i>Flowchart</i> pengerjaan tugas akhir.....	20
Gambar 3.2	<i>Airfoil</i> turbin angin.....	22
Gambar 3.3	<i>Gear</i> penghubung antara motor servo dengan <i>blade</i>	22
Gambar 3.4	Servo motor tipe HS-805BB.....	23
Gambar 3.5	Penyangga <i>blade</i> dan motor servo.....	23
Gambar 3.6	Rangkaian <i>rotary encoder</i>	25
Gambar 3.7	<i>Prototype</i> turbin angin.....	25
Gambar 3.8	Susunan kipas angin untuk uji sistem.....	26
Gambar 3.9	Blok diagram sistem kontrol <i>Pitch</i> turbin angin.....	27
Gambar 3.10	Diagram alir optimalisasi menggunakan PSO.....	29
Gambar 4.1	Pengujian sistem turbin angin dengan perubahan sudut 4^0 pada kecepatan angin 2,6 m/s.....	38
Gambar 4.2	Pengujian sistem turbin angin dengan perubahan sudut 4^0 pada kecepatan angin 1,9 m/s.....	38
Gambar 4.3	Pengujian sistem turbin angin dengan perubahan sudut 4^0 pada kecepatan angin 1,6 m/s.....	39
Gambar 4.4	Simulink uji sistem pemodelan secara <i>open loop</i>	42
Gambar 4.5	Kurva hasil uji <i>open loop</i> sistem turbin angin <i>input</i> sudut <i>pitch</i> 4^0	42

Gambar 4.6	Diagram simulink sistem <i>close loop</i> turbin angin.....	43
Gambar 4.7	Grafik penurunan nilai IAE terhadap jumlah Iterasi.....	44
Gambar 4.8	Pengujian <i>set point</i> 40 rpm.....	45
Gambar 4.9	Pengujian <i>set point</i> 60 rpm.....	46
Gambar 4.10	Pengujian <i>set point</i> 80 rpm.....	46
Gambar 4.11	Pengujian <i>set point</i> 100 rpm.....	47
Gambar 4.12	Pengujian <i>set point</i> 40 rpm metode PI-Fuzzy...	48
Gambar 4.13	Pengujian <i>set point</i> 60 rpm metode PI-Fuzzy...	49
Gambar 4.14	Pengujian <i>set point</i> 80 rpm metode PI-Fuzzy...	49
Gambar 4.15	<i>Tracking set point</i> 40-80-40 rpm.....	50

DAFTAR TABEL

Tabel 3.1	Variasi Mode Kipas Angin Sebagai Sumber Angin.....	26
Tabel 4.1	Hasil Pengujian Sensor <i>Rotary Encoder</i>	33
Tabel 4.2	Pengujian Sudut <i>Pitch</i> Maksimum Sistem Turbin Angin.....	34
Tabel 4.3	Hasil Pengujian Nilai C_p Turbin Angin.....	37
Tabel 4.4	Interpolasi Waktu untuk Respon 63% <i>Steady</i>	40
Tabel 4.5	Interpolasi Waktu untuk Respon 28% <i>Steady</i>	41
Tabel 4.6	<i>Error</i> Pengujian Input 4^0	43
Tabel 4.7	Indeks Performansi Hasil Pengujian <i>Set Point</i>	48
Tabel 4.8	Perbandingan Kriteria Performansi Hasil Tuning Parameter PID Menggunakan PSO dan Fuzzy.....	50
Tabel C1	Respon <i>Real Plant</i> Turbin Angin Sumber Kecepatan	

Halaman ini memang dikosongkan

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kecepatan angin di Indonesia terpisah menjadi tiga kategori mulai dari *Zone I* dengan kecepatan angin rata-rata sebesar 2,5 – 4,0 m/s, *Zone II* dengan kecepatan angin rata-rata sebesar 4,0 – 5,0 m/s serta *Zone III* dengan kecepatan angin rata-rata sebesar 5,0 m/s ke atas. Ketiga kategori kecepatan angin tersebut tergolong kecepatan angin rendah. Meskipun demikian, sumber energi angin tersebut perlu dimanfaatkan mengingat akan terjadinya krisis energi pada masa mendatang apabila sumber energi fosil telah habis digunakan, karena berdasarkan data dari Badan Geologi Kementerian ESDM, cadangan batu bara di Indonesia akan habis di antara tahun 2033 dan 2036 (Musyafa', 2016). Teknologi dalam pemanfaatan sumber energi baru dan terbarukan berupa angin ini adalah teknologi Sistem Konversi Energi Angin (SKEA). Perkembangan teknologi SKEA di Indonesia telah dimulai semenjak tahun 1980-an, hal tersebut diungkapkan oleh Badan Penelitian dan Pengembangan (LITBANG) tentang teknologi SKEA. Hingga akhir tahun 2011, total kapasitas energi listrik yang dihasilkan dari teknologi ini sebesar 1,4 MW. Nilai tersebut, menempatkan Indonesia pada peringkat ke-70 dari 83 negara yang ikut berpartisipasi dalam pemanfaatan energi angin melalui teknologi SKEA (Musyafa', 2016). Dari data tersebut, menunjukkan bahwa perkembangan teknologi SKEA ini masih perlu untuk dikembangkan lagi.

Blade pada turbin angin bergerak dikarenakan adanya gaya *lift* dan gaya *drag* yang dihasilkan. Pergerakan *blade* turbin angin kemudian dikonversi menjadi energi listrik. Selama proses konversi energi gerak angin menjadi energi listrik, tidak dipungkiri terdapat *loss energy* atau energi yang hilang. Sehingga dibutuhkan teknik optimalisasi pada turbin angin sehingga *loss energy* dapat dikurangi semaksimal mungkin. Salah satu metode optimalisasi yang dapat digunakan adalah dengan menerapkan penggunaan sudut *pitch* pada bilah turbin yang paling optimal

untuk setiap masukan angin dengan kecepatan tertentu. Hal ini telah dibuktikan pada penelitian sebelumnya mengenai pemilahan sudut *pitch* optimal pada blade turbin untuk setiap masukan dengan kecepatan angin yang berbeda (Musyafa', 2011). Harapan dari hasil penelitian tersebut adalah dengan dilakukannya metode optimalisasi energi listrik pada turbin angin dengan menentukan ketepatan sudut kemiringan pada bilah turbin. Langkah selanjutnya adalah menyusun sebuah kontroller PID untuk mengatur agar keluaran daya pada turbin angin terjaga dalam kondisi optimal (Sunarto, 2011).

Fokus dari penelitian ini yaitu perancangan *prototype* turbin angin. Hasil dari perancangan selanjutnya dicari sudut *pitch* optimalnya pada setiap kecepatan angin yang diterima. Langkah terakhir adalah merancang sistem kontrol sudut *pitch* pada turbin angin dengan pencarian parameter kontrollernya menggunakan metode PSO (*Particle Swarm Optimization*) dan Arduino sebagai prosesornya.

1.2 Perumusan Masalah

Dari latar belakang yang telah diuraikan, maka permasalahan yang diangkat pada penelitian ini adalah sebagai berikut:

- a. Bagaimana merancang sistem turbin angin dengan sumber kecepatan angin yang relatif rendah.
- b. Bagaimana merancang sebuah sistem kontrol sudut *pitch* untuk diterapkan pada turbin angin dengan metode optimalisasi PSO serta bagaimana performansinya.

1.3 Tujuan Tugas Akhir

Tujuan dari penelitian ini berdasarkan permasalahan yang diuraikan, adalah sebagai berikut ini:

- a. Merancang sistem turbin angin dengan sumber kecepatan angin yang relatif rendah.
- b. Mendapatkan sebuah sistem kontrol sudut *pitch* untuk diterapkan pada turbin angin dengan metode optimalisasi PSO serta diketahui performansinya.

1.4 Batasan Masalah

Batasan masalah dalam tugas akhir ini antara lain :

- a. Penampang bilah turbin angin menggunakan tipe *airfoil* NREL S835 pada bagian pangkal, S833 pada bagian tengah serta S834 untuk bagian ujung bilah.
- b. Kinerja dari generator dianalogikan dengan jumlah putaran yang dihasilkan turbin angin setiap menit (rpm).
- c. Pemodelan matematis turbin angin didekatkan menggunakan metode FOPDT (*First Orde Plus Dead Time*) yang bekerja dengan *input* kecepatan angin sebesar 2,6 m/s.
- d. Sistem controller menggunakan kontrol PID dengan pencarian parameter controller menggunakan metode PSO (*Particle Swarm Optimization*) serta mikroprosesor yang digunakan adalah Arduino Uno.

Halaman ini memang dikosongkan

BAB II

TINJAUAN PUSTAKA

Pada bab ini, berisi pustaka tentang segala materi yang berkaitan dengan penelitian yang dikerjakan, sehingga nantinya dapat mengarahkan pola berpikir dalam menemukan solusi permasalahan yang dalam penelitian ini. Tinjauan pustaka didapatkan melalui beberapa referensi baik dalam jenis *text book*, *manual book* maupun jurnal ilmiah yang sebelumnya telah dipublikasikan.

2.1 Turbin Angin

Turbin angin adalah kincir angin yang digunakan untuk membangkitkan tenaga listrik yang berasal dari energi kinetik angin. Energi kinetik angin tersebut menyentuh penampang bilah turbin angin yang kemudian bilah kincir angin akan bergerak pada porosnya. Persamaan umum energi kinetik angin adalah (da Rosa, 2009) :

$$E_k = \frac{1}{2} mv^2 \quad (2.1)$$

Dimana,

E_k = Energi kinetik angin

m = massa angin yang mengalir

v = kecepatan angin

Angin (fluida) dengan jumlah konsentrasi molekul tertentu (n) yang bergerak melewati suatu penampang dengan kecepatan aliran tertentu, disebut dengan flux (ϕ). (da Rosa, 2009)

$$\phi = n \cdot v \text{ (m}^{-2} \cdot \text{s}^{-1}\text{)} \quad (2.2)$$

Sedangkan daya yang dibawa oleh suatu angin dirumuskan sebagai berikut (da Rosa, 2009) :

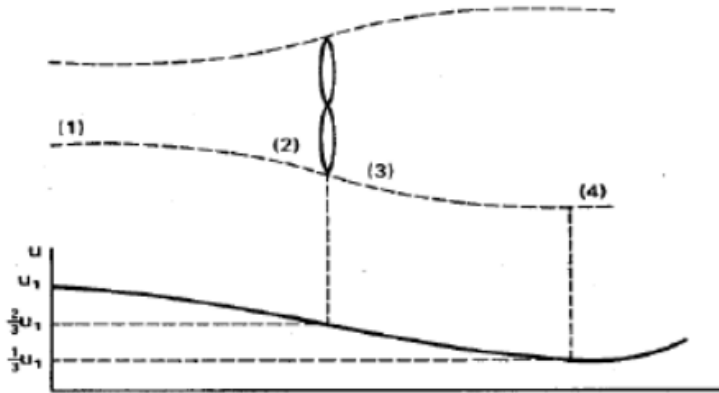
$$P_w = E_k \phi$$

$$P_w = \frac{1}{2} \rho A v^3$$

$$P_w = \frac{1}{2} \rho v^3 \quad (\text{W m}^{-2}) \quad (2.3)$$

Dimana, ρ atau massa jenis udara merupakan perkalian antara massa dengan konsentrasi udara tersebut.

Besarnya daya angin yang diekstrak oleh sebuah turbin angin, diperlihatkan Gambar 2.1 mengenai *tube* angin yang mengenai turbin angin.



Gambar 2.1 *Tube* angin yang melewati turbin angin (Musyafa', 2016)

Daya angin berasal dari sumber (titik 1) melewati turbin angin dengan luas sapuan (A) yang bergantung pada jari-jari turbin angin. Berikut ini, persamaan daya trbin angin ditunjukkan oleh persamaan berikut (da Rosa, 2009) :

$$P_A = \frac{1}{2} \rho v^3 C_P \quad (2.4)$$

Dimana, C_P merupakan nilai *Power Coefficien*.

Untuk mengetahui nilai *Power Coefficient* (C_p), maka dapat dilakukan dengan cara membandingkan Daya angin sebelum diekstrak oleh turbin angin dengan daya angin setelah turbin angin. Berikut ini persamaannya (da Rosa, 2009) :

$$\begin{aligned} C_p &= P / P_w \\ C_p &= 0.25 \rho A (v_1 + v_4) (v_1^2 - v_4^2) / 0.5 \rho A v_1^3 \\ C_p &= 0.5 (1 + v_4/v_1) (1 - (v_4/v_1)^2) \end{aligned} \quad (2.5)$$

Dimana,

v_1 = Kecepatan angin sebelum melewati turbin angin

v_4 = Kecepatan angin setelah melewati turbin angin

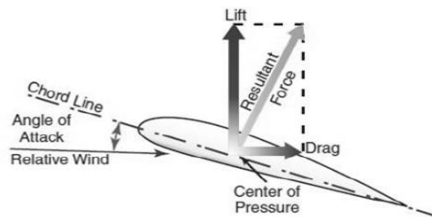
Pada aturan Betz, rasio antara kecepatan angin sebelum dan setelah melewati turbin angin (v_4/v_1) maksimum sebesar 1/3. Sehingga, nilai C_p maksimum sebesar (da Rosa, 2009) :

$$\begin{aligned} C_{p_{\max}} &= 0.5 (1 + 1/3) (1 - (1/3)^2) \\ C_{p_{\max}} &= 16/27 = 0.59259259 \end{aligned}$$

Nilai 0.59259259 kemudian ditetapkan sebagai Betz's Limit.

2.1.1 Gaya Aerodinamik pada Turbin Angin

Sudu atau *rotor* berfungsi untuk menghasilkan putaran akibat gaya angin dan menggerakkan poros turbin dan poros generator yang kemudian akan menghasilkan energi listrik. Jari-jari sudu adalah jarak sudu dari permukaan poros *rotor* sampai ujung dari sudu.



Gambar 2.2 Fenomena *Drag* dan *Lift* (Sunarto, 2014)

Pada sudu turbin angin akan terjadi tegangan geser pada permukaannya ketika kontak dengan udara. Distribusi tegangan geser pada permukaan sudu ini dipresentasi dengan adanya gaya tekan (*drag*) yang arahnya sejajar dengan arah aliran fluida dan gaya angkat (*lift*) yang arahnya tegak lurus dari arah aliran fluida. Kedua gaya ini menyebabkan sudu dapat berputar. Kedua gaya ini dipengaruhi oleh bentuk sudu, luas permukaan bidang sentuh, sudut serang, dan kecepatan angin. Fenomena *drag* dan *lift* diilustrasikan pada Gambar 2.2.

Untuk mempermudah perhitungan fenomena *drag* dan *lift*, maka dengan metode numerik, diperkenalkan *drag coefficient* dan *lift coefficient* (koefisien gaya hambat dan koefisien gaya angkat) yang dilambangkan dengan C_D dan C_L . Besarnya C_L dan C_D bergantung dari bentuk melintang sudu yang digunakan dan sudut serang (α). Secara matematis, hubungan gaya *drag* dan *lift* dengan koefisiennya dapat dirumuskan sebagai berikut (Sunarto, 2014) :

$$F_D = \frac{1}{2} C_D \rho v^2 A \quad (2.6)$$

$$F_L = \frac{1}{2} C_L \rho v^2 A \quad (2.7)$$

Dimana,

- ρ = massa jenis udara
- A = luas penampang sudu
- v = adalah kecepatan angin.

Teori momentum sederhana (teori Betz's) menunjukkan persamaan dasar untuk menghitung besarnya output daya mekanik dari rotor (*mechanical power output*). Dengan menggunakan *power coefficient* C_P , daya rotor dapat dihitung sebagai fungsi dari kecepatan angin (Sunarto, 2014).

$$P_R = C_P \rho / 2 v_w^3 A \quad (2.8)$$

Dimana,

- A = luas sudu
- v_w = kecepatan angin
- C_P = koefisien daya rotor
- ρ = massa jenis udara

P_R = daya rotor

Power coefficient merupakan fungsi dari besaran nilai *tip speed ratio* (λ) dan sudut *pitch* (θ). *Tip speed ratio* (λ), yang secara umum rumuskan menjadi (Sunarto, 2014) :

$$\text{Tip speed ratio, } \lambda = \omega R / v_w \quad (2.9)$$

Dimana,

ω = kecepatan sudut
 v_w = kecepatan angin
 R = jari-jari sudu

Nilai *tip speed ratio* (λ) merupakan nilai konstan, sehingga nilai C_p akan bergantung pada nilai sudut *pitch*.

2.1.2 Pemilihan Jenis *Airfoil*

Badan laboratorium nasional NREL (*National Renewable Energy Laboratory*) telah melakukan penelitian dan merilis beberapa *airfoil* yang cocok untuk diterapkan pada turbin angin *horizontal axis* dengan skala kecil hingga besar. Untuk turbin angin dengan diameter rotor 1 – 3 meter, NREL mengeluarkan seri S822 dan S823 yang kemudian digantikan oleh seri S83n (S833, S834, S835). *Airfoil* seri S83n merupakan *airfoil* tebal yang cocok digunakan untuk turbin angin dengan variabel *speed* dan variabel *pitch* dengan tipikal *foil* rendah *noise* dan koefisien *lift* yang tinggi. Berdasarkan desain spesifikasi tersebut, maka ketiga *airfoil* tersebut dapat digunakan pada turbin angin dengan diameter 2 m yang dilengkapi dengan variabel *pitch control*. Untuk bagian pangkal digunakan tipe S835, bagian tengah tipe S833, dan bagian ujung bilah tipe S834.

2.2 Komponen Turbin Angin

Pada turbin angin yang dirancang, beberapa komponen utama yang diperlukan selain *blade* turbin angin, terdapat juga komponen pendukung seperti motor servo sebagai penggerak *blade*, *rotational connector* (*slipring connector*) sebagai penghubung antara kabel dari komponen turbin angin yang

berputar dengan komponen turbin angin yang tidak berputar, *rotary encoder* sebagai sensor rpm serta modul Arduino Uno sebagai prosesor sistem turbin angin. Berikut ini penjelasan dari mekanisme setiap komponen pendukung tersebut.

2.2.1 Motor Servo

Motor servo merupakan motor yang mampu bekerja dua arah (CW dan CCW) dimana arah dan sudut pergerakan rotornya dapat dikendalikan hanya dengan memberikan pengaturan *duty cycle* sinyal PWM pada bagian pin kontrolnya (Sigit, 2007). Contoh motor servo yakni Futaba yang diperlihatkan oleh Gambar 2.3.



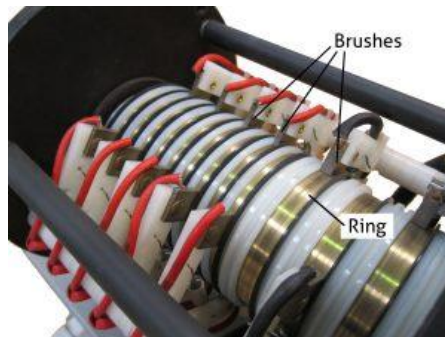
Gambar 2.3 Bentuk fisik motor servo (Sigit, 2007)

Terdapat dua jenis motor servo diantaranya jenis standar, dimana motor servo jenis ini hanya mampu bergerak dua arah (CW dan CCW) dengan defleksi masing-masing sudut mencapai 90° sehingga total defleksi sudut dari kanan – tengah – kiri adalah 180° . Sedangkan jenis lainnya dari motor servo adalah jenis *Continuous*, dimana motor servo jenis ini mampu bergerak dua arah (CW dan CCW) tanpa batasan defleksi sudut putar (dapat berputar secara kontinyu) (Sigit, 2007).

Terdapat 3 *input* sinyal motor servo yakni sinyal Vcc, Gnd serta data. Besarnya nilai Vcc setiap motor servo berbeda-beda. Nilai tersebut tergantung pada setiap spesifikasi motor servo. Sedangkan nilai sinyal data bergantung pada sudut yang ingin digerakkan.

2.2.2 Rotational Connector

Rotational Connector atau sering disebut dengan *Slip Ring connector* merupakan perangkat yang mampu mentransfer daya atau sinyal di selama terjadi putaran. Pada dasarnya, *slip ring* terdiri dari dua komponen utama - sikat logam, dan cincin logam. Sikat berada dalam kontak dengan cincin logam yang berputar. Hasil dari gesekan, menyebabkan daya listrik dapat dihantarkan (Powerbyproxi, 2016). Mekanisme *Slip ring* ditunjukkan oleh Gambar 2.4.

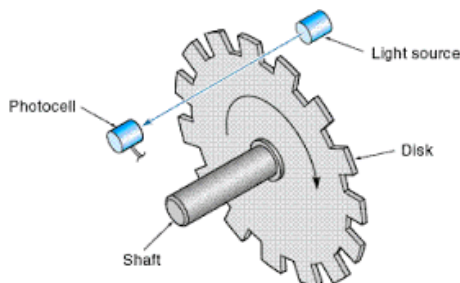


Gambar 2.4 Mekanisme *Slip Ring* (Powerbyproxi, 2016)

Sikat (*Brushes*) adalah kontak stasioner terbuat dari grafit atau logam, yang kemudian menggosok terhadap bagian luar cincin metal yang berputar. Terbuat dari serat logam, sikat ini yang membuat kontak fisik dengan cincin. Cincin (*Ring*) merupakan komponen yang terus berputar mengikuti putaran *rotor* turbin. Jumlah *ring* ditemukan dalam *slip ring* yang diberikan dapat bervariasi tergantung pada kebutuhan perangkat, diletakkan sejajar sepanjang sumbu poros. Isolasi antara poros dan cincin umumnya terbuat dari sejumlah bahan sintesis, termasuk nilon dan plastik fenolik (Powerbyproxi, 2016).

2.2.3 Rotary Encoder

Encoder merupakan perangkat elektromekanik yang mampu mengukur perpindahan. *Rotary Encoder* memiliki arti sebuah perangkat yang digunakan untuk mengukur perpindahan suatu benda yang berputar. Desain mekanik dari sebuah *rotary encoder* ditunjukkan oleh Gambar 2.5.

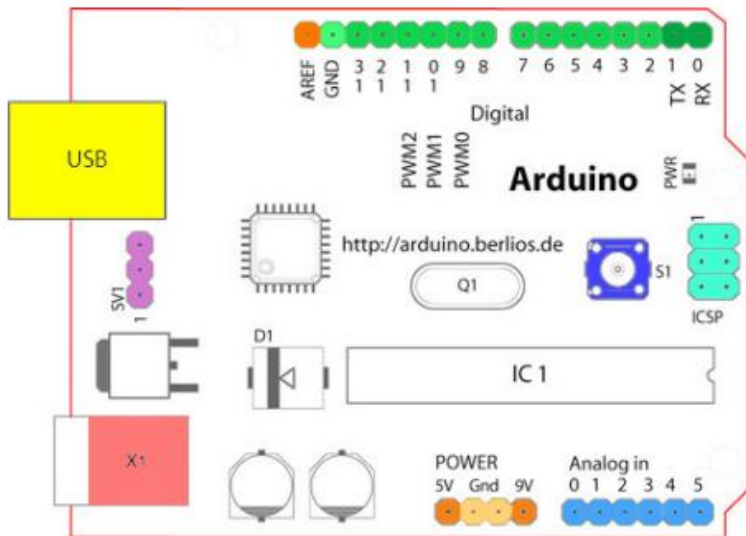


Gambar 2.5 Desain Umum *Rotary Encoder* (Andrade, 2016)

Prinsip kerja dari sensor ini, poros yang berputar (*shaft*) disambungkan dengan sebuah piringan (*disk*) yang memiliki lubang yang tersebar merata. Saat rangkaian sumber cahaya (*light source*) diberikan, cahaya masuk pada *photodiode (photocell)* yang tidak terhalang piringan (*disk*), maka akan menghasilkan tegangan, begitu juga sebaliknya saat terhalangi maka akan menghasilkan tegangan sekitar berbeda pula. Perbedaan tegangan tersebut dianalogikan sebagai sinyal *biner* pada perangkat mikrokontroler. Pada perangkat tersebut, diketahui pula waktu perpindahan antara sinyal “0” dan “1”.

2.2.4 Arduino

Untuk mengatur sudut *pitch* dari bilah turbin angin maka diperlukan suatu mekanisme elektronik untuk memudahkan pengaturan sudut sekaligus untuk mencatat kecepatan putar turbin angin dan kecepatan angin pada kecepatan angin tertentu. Pada tugas akhir ini digunakan Arduino sebagai mikroprosesor untuk mengintegrasikan sensor-sensor yang terdapat pada turbin angin (Devika, 2014).



Gambar 2.6 Papan Arduino (Devika, 2014)

Arduino adalah pengendali mikro *single-board* yang bersifat *open-source*, yang dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang. *Hardware*-nya memiliki prosesor Atmel AVR (mikrokontroler AVR 8 bit dan ARM 32 bit.) dan *software*-nya memiliki bahasa pemrograman sendiri. *Open source* IDE yang digunakan untuk membuat aplikasi mikrokontroler yang berbasis *platform* arduino. Mikrokontroler sendiri adalah komputer berskala kecil yang terdiri atas mikroprosesor, memori, dan beberapa fitur terintegrasi dalam satu *Integrated Circuit* (IC). Salah satu keunggulan dari Arduino adalah tidak perlu perangkat *chip programmer* karena di dalamnya sudah ada *boot loader* yang akan menangani *upload* program dari komputer dan sudah memiliki sarana komunikasi USB, sehingga pengguna laptop yang tidak memiliki *port serial*/RS323 bisa menggunakannya (Devika, 2014).

Dengan mengambil contoh sebuah papan Arduino tipe USB, bagian-bagiannya dapat dijelaskan pada Gambar 2.6.

a. 14 pin *input/output* digital (0-13)

Berfungsi sebagai *input* atau *output*, dapat diatur oleh program. Khusus untuk 6 buah pin 3, 5, 6, 9, 10 dan 11, dapat juga berfungsi sebagai pin analog *output* dimana tegangan *output*-nya dapat diatur. Nilai sebuah pin *output* analog dapat diprogram antara 0 – 255, dimana hal itu mewakili nilai tegangan 0 – 5V.

b. USB

Berfungsi untuk memuat program dari komputer ke dalam papan, komunikasi serial antara papan dan komputer, memberi daya listrik kepada papan

c. Sambungan SV1

Sambungan atau *jumper* untuk memilih sumber daya papan, apakah dari sumber eksternal atau menggunakan USB. Sambungan ini tidak diperlukan lagi pada papan Arduino versi terakhir karena pemilihan sumber daya eksternal atau USB dilakukan secara otomatis.

d. Q1 – Kristal (*quartz crystal oscillator*)

Jika mikrokontroler dianggap sebagai sebuah otak, maka kristal adalah jantung-nya karena komponen ini menghasilkan detak-detak yang dikirim kepada mikrokontroler agar melakukan sebuah operasi untuk setiap detak-nya. Kristal ini dipilih yang berdetak 16 juta kali per detik (16MHz).

e. Tombol *Reset* S1

Untuk me-*reset* papan sehingga program akan mulai lagi dari awal. Perhatikan bahwa tombol *reset* ini bukan untuk menghapus program atau mengosongkan mikrokontroler.

f. *In – Circuit Serial Programming* (ICSP)

Port ICSP memungkinkan pengguna untuk memprogram mikrokontroler secara langsung, tanpa melalui *bootloader*. Umumnya pengguna Arduino tidak melakukan ini sehingga ICSP tidak terlalu dipakai walaupun disediakan.

g. IC 1 = Mikrokontroler Atmega

Komponen utama dari papan Arduino, di dalamnya terdapat CPU, ROM dan RAM.

h. X1 = Sumber Daya Eksternal

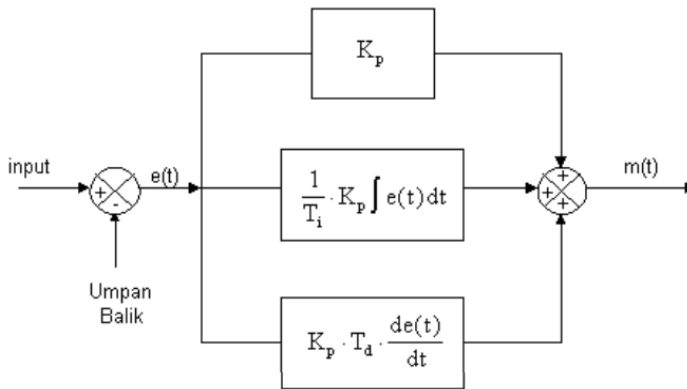
Jika hendak disuplai dengan sumber daya eksternal, papan Arduino dapat diberikan tegangan dc antara 9-12V.

i. 6 Pin *Input* analog (0-5)

Pin ini sangat berguna untuk membaca tegangan yang dihasilkan oleh sensor analog, seperti sensor suhu. Program dapat membaca nilai sebuah pin *input* antara 0 – 1023, dimana hal itu mewakili nilai tegangan 0 – 5V.

2.3 Kontroler PID (*Proportional Integral Derivative*)

Didalam suatu sistem kontrol kita mengenal adanya beberapa macam aksi kontrol, diantaranya yaitu aksi kontrol Proporsional, aksi kontrol Integral dan aksi kontrol *Derivative*. Masing-masing aksi kontrol ini mempunyai keunggulan-keunggulan tertentu, dimana aksi kontrol proporsional mempunyai keunggulan *rise time* yang cepat, aksi kontrol Integral mempunyai keunggulan untuk memperkecil *error*, dan aksi kontrol *derivative* mempunyai keunggulan untuk memperkecil *error* atau meredam *overshot/undershot*.



Gambar 2.7 Blok Diagram Pengendali PID (Susanto)

Suatu sistem dapat menghasilkan *output* dengan *risetime* yang cepat dan *error* yang kecil kita dapat menggabungkan ketiga aksi kontrol ini menjadi aksi kontrol PID. Diagram blok dari sistem kendali PID ditunjukkan oleh Gambar 2.7. Sistem kontrol PID merupakan gabungan metode pengaturan yang terdiri dari kendali Proportional, Integral, dan *Derivative*.

2.4 PSO (*Particle Swarm Optimzation*)

Particle Swarm Optimization (PSO) adalah salah satu teknik optimasi dan termasuk jenis teknik komputasi evolusi yang dikembangkan oleh Dr. Eberhart dan Dr. Kennedy pada tahun 1995. Jauh lebih modern dibandingkan dengan metode *artificial intelligence* yang lain seperti metode fuzzy yang diperkenalkan pada tahun 1965. Metode ini memiliki sifat *robust* yang bagus untuk memecahkan persoalan yang mempunyai karakteristik *nonlinear* dan *nondifferentiability*, *multiple optima*, dimensi besar melalui adaptasi yang diturunkan dari teori *psychology*-sosial. PSO terinspirasi dari perilaku gerakan kawanan hewan seperti ikan (*school of fish*), hewan herbivor (herd), dan burung (flock) yang kemudian tiap objek hewan disederhanakan menjadi sebuah partikel. Suatu partikel dalam ruang memiliki posisi yang dikodekan sebagai vektor koordinat. Vektor posisi ini dianggap sebagai keadaan yang sedang ditempati oleh suatu partikel di ruang pencarian. Setiap posisi dalam ruang pencarian merupakan alternatif solusi yang dapat dievaluasi menggunakan fungsi objektif. Setiap partikel bergerak dengan kecepatan v . Ciri khas dari PSO adalah pengaturan kecepatan partikel secara heuristik dan probabilistik. Jika suatu partikel memiliki kecepatan yang konstan maka jika jejak posisi suatu partikel divisualisasikan akan membentuk garis lurus. Dengan adanya faktor eksternal yang membelokkan garis tersebut yang kemudian menggerakkan partikel dalam ruang pencarian maka diharapkan partikel dapat mengarah, mendekati, dan pada akhirnya mencapai titik optimal. faktor eksternal yang dimaksud antara lain posisi terbaik yang pernah dikunjungi suatu partikel, posisi terbaik seluruh partikel (diasumsikan setiap partikel mengetahui posisi terbaik setiap partikel lainnya), serta faktor kreativitas untuk melakukan

eksplorasi. PSO memiliki kesamaan sifat dengan teknik komputasi seperti Algoritma Genetika. Sistem PSO diinisialisasi oleh sebuah populasi solusi secara acak dan selanjutnya mencari titik optimum dengan cara meng-*update* tiap hasil pembangkitan. Metode optimasi yang didasarkan pada *swarm intelligence* ini disebut algoritma *behaviorally inspired* sebagai alternatif dari algoritma genetika, yang sering disebut *evolution-based procedures*. Dalam konteks optimasi multivariabel, kawanan diasumsikan mempunyai ukuran tertentu atau tetap dengan setiap partikel posisi awalnya terletak di suatu lokasi yang acak dalam ruang multidimensi. Setiap partikel diasumsikan memiliki dua karakteristik: posisi dan kecepatan. Setiap partikel bergerak dalam ruang / *space* tertentu dan mengingat posisi terbaik yang pernah dilalui atau ditemukan terhadap sumber makanan atau nilai fungsi objektif. Setiap partikel menyampaikan informasi atau posisi bagusnya kepada partikel yang lain dan menyesuaikan posisi dan kecepatan masing-masing berdasarkan informasi yang diterima mengenai posisi yang bagus tersebut. Sebagai contoh, misalnya perilaku burung-burung dalam kawanan burung. Meskipun setiap burung mempunyai keterbatasan dalam hal kecerdasan, biasanya ia akan mengikuti kebiasaan (*rule*) seperti berikut:

- Seekor burung tidak berada terlalu dekat dengan burung yang lain.
- Burung tersebut akan mengarahkan terbangnya ke arah rata-rata keseluruhan burung.
- Akan memposisikan diri dengan rata-rata posisi burung yang lain dengan menjaga sehingga jarak antar burung dalam kawanan itu tidak terlalu jauh.

Jadi, PSO dikembangkan dengan berdasarkan pada model berikut:

- Ketika seekor burung mendekati target atau makanan akan secara cepat mengirim informasi kepada burung-burung yang lain dalam kawanan tertentu.
- Burung yang lain akan mengikuti arah menuju ke makanan tetapi tidak secara langsung.

- Ada komponen yang tergantung pada pikiran setiap burung, yaitu memorinya tentang apa yang sudah dilewati pada waktu sebelumnya.

Model ini akan disimulasikan dalam ruang dengan dimensi tertentu dengan sejumlah iterasi sehingga di setiap iterasi, posisi partikel akan semakin mengarah ke target yang dituju (minimasi atau maksimasi fungsi). Ini dilakukan hingga maksimum iterasi dicapai atau bisa juga digunakan kriteria penghentian yang lain. Secara matematis deskripsi di atas ditampilkan sebagai berikut (Karyatulisilmiah, 2005) :

$$v[i] = v[i] + c1 * rand() * (pbest[i] - sekarang[i]) + c2 * rand() * (gbest[i] - sekarang[i]) \quad (2.10)$$

$$sekarang[i] = persent[i] + v[i] \quad (2.11)$$

Dimana:

$v[i]$ adalah kecepatan partikel

$persent[i]$ adalah partikel saat ini (solusi)

$pbest[i]$ dan $gbest[i]$ adalah posisi terbaik lokal dan global

$rand()$ adalah nomor acak antara (0,1)

$c1, c2$ faktor belajar, biasanya $c1 = c2 = 2$

Metode PSO memiliki beberapa keunggulan, yaitu implementasinya mudah, konvergensinya stabil, algoritmanya sederhana dan efisien dalam komputasi. Pada sebuah penelitian yang dilakukan oleh Alrijadjis, Sistem dengan *time delay* dimodelkan dalam *Simulink* dan algoritma PSO diimplementasikan dalam MATLAB. Hasil simulasi menunjukkan bahwa metode baru tuning menggunakan PSO memiliki performansi yang lebih baik dibandingkan dengan metode konvensional lainnya (Ziegler-Nichols atau ZN, *Internal Model Controller* atau IMC dan *Iterative Feedback Tuning* atau IFT). (Alrijadjis).

BAB III METODOLOGI PENELITIAN

3.1 Prosedur Umum Penelitian

Pada sub bab ini, dijelaskan segala prosedur dalam penelitian yang dilakukan sehingga tujuan penelitian tercapai. Prosedur tersebut digambarkan melalui *flowchart* yang ditunjukkan oleh Gambar 3.1. Berikut ini uraian dari beberapa tahapan dalam perancangan turbin angin serta kontroler sudut *pitch* turbin angin menggunakan metode PSO (*Particle Swarm Optimization*).

a. Perancangan Sistem Turbin Angin

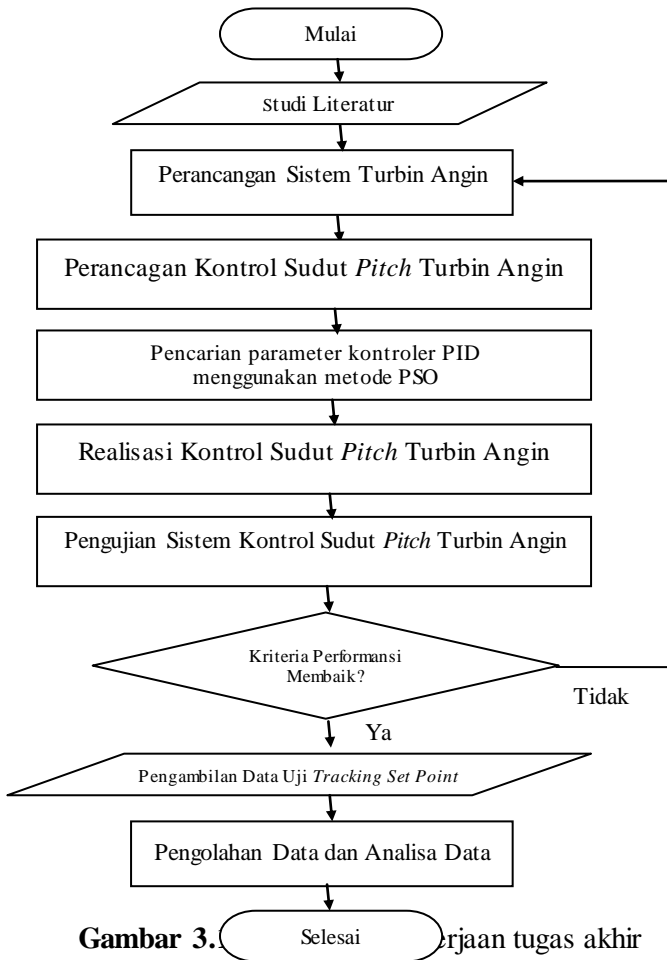
Perancangan yang dimaksud merupakan perancangan secara *hardware*. Perancangan tersebut dengan menerapkan beberapa spesifikasi turbin angin untuk sumber angin skala kecil. Setiap komponen turbin angin juga dilakukan pengujian terlebih dahulu kemudian dirangkai dan dilakukan pengujian secara keseluruhan. Selanjutnya, turbin angin tersebut diuji dengan mencari nilai sudut optimal pada beberapa kecepatan angin yang diberikan serta dilakukan *monitoring* kinerja turbin angin pada daerah sudut optimalnya.

b. Perancangan Kontrol Sudut Pitch Turbin Angin

Hasil dari pengujian dan *monitoring* alat secara *real* tersebut dicari pemodelan sistemnya kemudian dilakukan pengujian apakah pemodelan tersebut telah mewakili *plant* yang sebenarnya. Lalu dirancang pula sistem dengan kontroler PID untuk sistem turbin angin.

c. Pencarian Parameter Kontroler PID menggunakan PSO

Tahap pencarian parameter kontroler PID dilakukan menggunakan metode PSO. Dalam proses optimalisasi, dibutuhkan beberapa parameter optimasi PSO seperti penentuan jumlah individu dalam suatu populasi, penentuan nilai posisi, kecepatan, penentuan jumlah iterasi serta penentuan nilai bobot inersia. Hasil pencarian nilai parameter PID kemudian diterapkan pada *real plant* sistem turbin angin.



d. Realisasi Kontrol Sudut *Pitch* Turbin Angin

Nilai parameter kontroler kemudian diaplikasikan pada *hardware* turbin angin dengan menggunakan prosesor Arduino Uno. Pada tahap ini, merupakan tahap penggabungan antara *software* dan juga *hardware* pada turbin angin. Kontrol sudut

pitch turbin angin diaplikasikan dalam bahasa pemrograman berupa bahasa C untuk menjalankan prosesor Arduino Uno. Dari perancangan ini, merupakan penentu apakah performansi turbin angin membaik daripada saat sebelum diberikan kontrol PID. Parameter tersebut berupa *settling time*, *rise time* serta *maksimum overshoot*.

e. Pengujian Sistem Kontrol Sudut *Pitch* Turbin Angin

Pada pengujian ini merupakan tahap pengujian *tracking set point* untuk sistem kontrol sudut *pitch* turbin angin secara *real plant*. *Setpoint* diberikan secara bervariasi dengan batasan rpm maksimum yang dapat dicapai oleh turbin angin tersebut. Parameter

f. Pengolahan dan Analisa Data

Seluruh data hasil pengujian maupun data spesifikasi turbin angin kemudian diolah serta dilakukan analisa setiap performansinya. Hasil pengolahan data tersebut kemudian dianalisa kelebihan dan kekurangannya.

3.2 Perancangan Turbin Angin

3.2.1 Pemasangan Komponen Turbin Angin

Komponen penyusun yang digunakan dalam pembuatan turbin angin antara lain *blade*, motor servo, sensor *rotary encoder*, penopang *blade*, *rotational connector*, tiang penyangga, *main plate*, serta modul Arduino Uno. Berikut ini penjelasan mengenai spesifikasi komponen penyusun turbin angin.

Komponen pertama adalah *blade* turbin angin. Desain *blade* turbin angin yang digunakan menggunakan tipe S835 pada bagian bagian pangkal, tipe S833 pada bagian tengah dan tipe S834 pada bagian ujung bilah. Panjang *blade* turbin yang dibuat yakni sebesar 1,1 m. Penampang fisik *blade* turbin ditunjukkan oleh Gambar 3.2. Bahan penyusunnya menggunakan bahan jenis *fiber glass* yang kemudian dilapisi dengan cat sehingga bahan relatif ringan.



Gambar 3.2 Airfoil turbin angin

Jumlah *blade* yang digunakan dalam turbin angin ini sebanyak 3 buah. Ketiga *blade* dikondisikan dengan berat yang sama yakni seberat 990 gram. Pada pangkal *blade* disambungkan dengan *bearing* untuk memperhalus pergerakan *blade* saat digerakkan dengan motor servo serta di bagian ujungnya disambung dengan sebuah *gear*.

Gear pada *blade* lalu dihubungkan dengan *gear* pada motor servo. *Gear* motor servo sendiri memiliki diameter yang lebih kecil daripada diameter *gear* pada *blade*. Rasio perbandingan antara *gear* motor servo dibandingkan dengan *gear blade* sebesar 1,4 : 2. Ini artinya, saat motor servo bergerak satu putaran penuh, *blade* bergerak hanya 0,7 putaran atau tidak sampai satu putaran penuh. *Gear* penghubung antara motor servo dengan *blade* ditunjukkan oleh Gambar 3.3.



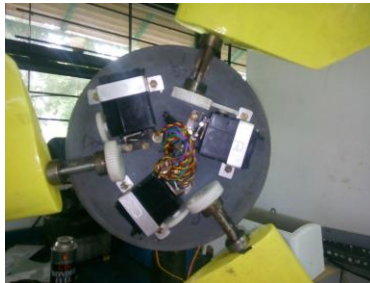
Gambar 3.3 Gear penghubung antara motor servo dengan *blade*

Motor servo yang digunakan merupakan motor servo dengan tipe HS-805BB. Tipe ini memiliki karakteristik mampu menggerakkan objek dengan torsi sebesar 19,8 Kg.cm dengan putaran setiap sudut 1° . Bentuk fisik dari motor servo untuk tipe ini ditunjukkan pada Gambar 3.4.



Gambar 3.4 Servo motor tipe HS-805BB

Setiap motor servo didesain untuk menggerakkan setiap *blade*, sehingga diperlukan 3 buah motor servo. Semua komponen *blade* serta motor servo diletakkan pada sebuah penopang yang berbahan dasar PVC seperti yang diperlihatkan oleh Gambar 3.5.



Gambar 3.5 Penyangga *blade* dan motor servo

Dimensi penopang tersebut sebesar 28 cm dengan tebal 1,5 cm. Pada penopang tersebut, di bagian tengahnya dikopel dengan sebuah besi yang dijadikan poros putaran turbin angin. Semua komponen yang dikaitkan pada penyangga PVC tersebut akan ikut berputar selama turbin angin bergerak.

Setiap kabel pada motor servo dihubungkan dengan *slipring connector*. Pada saat turbin angin berputar, maka komponen berupa servo motor dan juga *blade* turbin angin akan berputar pada porosnya. *Slipring connector* tersebut mampu menghubungkan antara kabel yang ikut berputar dengan kabel

yang tidak ikut berputar. Sehingga daya untuk servo motor tidak akan putus selama turbin angin bergerak.

Diperlukan lima buah *channel* pada *slipring connector* dalam perancangan alat ini. Diantaranya 1 *channel* untuk *input* Vcc motor servo, 1 *channel* untuk *input* Gnd motor servo serta 3 *channel* untuk masing-masing *input* data motor servo. *Input* Vcc dan juga Gnd pada ketiga servo dihubungkan secara paralel. Daya diberikan dengan menggunakan sumber daya PLN 220 Vac yang kemudian disearahkan agar menjadi tegangan sebesar 6 Vdc. Arus yang diberikan juga dijaga tetap sebesar 5 A. Sedangkan *input* data pada setiap servo diberikan melalui pin *output* dari Arduino Uno.

Selama poros turbin bergerak, diperlukan juga sebuah sensor pembacaan kecepatan putaran turbin angin. Dalam perancangan ini, digunakan sensor *rotary encoder* untuk mengetahui rpm (*Rotation Per Minute*) dari putaran turbin angin. Nilai rpm tersebut dianalogikan sebagai besarnya keluaran daya dari turbin angin saat diberi daya angin. Artinya, semakin besar rpm dari poros turbin, maka daya keluaran turbin angin juga besar.

Rotary encoder yang digunakan memiliki lubang sebanyak 16 lubang. Pada saat turbin angin berputar, maka piringan tersebut ikut berputar dan dilakukan pembacaan perputaran dengan mekanisme *interrupt*. Dengan mekanisme tersebut, dilakukan pengukuran besar rpm (Rotasi Per Menit) dari turbin angin. Untuk mengetahui kinerja *rotary encoder*, maka dilakukan pengujian *rotary encoder* menggunakan motor listrik dengan variasi tegangan masukan lalu nilai rpm dari *rotary encoder* diuji dengan nilai hasil pengukuran menggunakan Tachometer. Bentuk dari *Rotary Encoder* dapat dilihat pada Gambar 3.6.



Gambar 3.6 Rangkaian *rotary encoder*

Prosesor yang digunakan dalam perancangan ini adalah prosesor Arduino Uno. Pin yang digunakan meliputi pin Vcc, Gnd, pin 2 sebagai pin *interrupt* untuk sensor *rotary encoder* serta 3 buah pin *output* (pin 10, 11 dan 12) untuk data yang menggerakkan servo motor.



Gambar 3.7 *Prototype* turbin angin

Main plate (Centre plate) digunakan untuk menopang poros *blade* turbin, rangkaian daya, *optpcoupler* serta Arduino. Bahan yang digunakan adalah besi dengan diameter sebesar 40 cm dan tebal 0,2 cm. Keseluruhan komponen turbin angin tersebut kemudian ditopang oleh tiang penyangga. Bentuk keseluruhan dari *prototype* turbin angin diperlihatkan oleh Gambar 3.7.

3.2.2 Pengujian Turbin Angin Tanpa Kontroller

Pengujian dilakukan dengan memberikan 3 jenis kecepatan angin. Pada setiap kecepatan angin, diberikan variasi perputaran

sudut *pitch* pada *blade* turbin angin dengan interval sebesar $1,4^{\circ}$. Pengujian dilakukan hingga sudut dimana turbin angin sudah tidak mampu berputar meskipun diberikan sumber angin. Variasi kecepatan angin yang diberikan berturut-turut adalah 1,6 ; 1,7; 1,8; 1,9; 2,3 dan 2,6 m/s.

Pemasangan kipas angin diperlihatkan oleh Gambar 3.8. Pengaturan mode kecepatan angin sehingga menghasilkan variasi kecepatan angin untuk pengujian ditunjukkan oleh Tabel 3.1. Dalam hal ini, tujuan dari penggunaan 3 kipas angin adalah agar semua *blade* terkena sumber angin. Sehingga kinerja turbin angin menjadi lebih optimal, mengingat diameter *blade* yang cukup besar melebihi diameter sebuah kipas angin.



Gambar 3.8 Susunan kipas angin untuk uji sistem

Tabel 3.1 Variasi Mode Kipas Angin Sebagai Sumber Angin

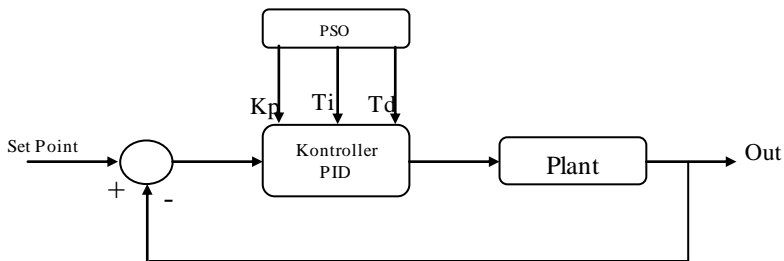
Mode	Kipas 1	Kipas 2	Kipas 3	Kecepatan rata-rata
1	<i>High</i>	<i>High</i>	<i>High</i>	2,6 m/s
2	<i>High</i>	<i>Low</i>	<i>High</i>	2,3 m/s
3	<i>Medium</i>	<i>Medium</i>	<i>Medium</i>	1,9 m/s
4	<i>Medium</i>	<i>Low</i>	<i>Medium</i>	1,8 m/s
5	<i>Low</i>	<i>High</i>	<i>Low</i>	1,7 m/s
6	<i>Low</i>	<i>Low</i>	<i>Low</i>	1,6 m/s

Jarak antara sumber angin dengan turbin angin adalah 3 m. Pada putaran turbin tertinggi, juga dilakukan pengujian *Power coefficient* (C_p) yakni dengan cara mengukur kecepatan angin di depan turbin angin (v_1) dan juga kecepatan angin di belakang turbin angin (v_2). Dari hasil pengukuran, kemudian dimasukkan

ke dalam persamaan (2.5). Hasil nilai C_p tersebut dibandingkan dengan nilai C_p *Betz Limit*.

3.3 Perancangan Kontrol *Pitch* Turbin Angin

Input dari sistem kontrol sudut *pitch* turbin angin ini merupakan nilai rpm yang dikehendaki. Kemudian, nilai tersebut dibandingkan dengan sinyal keluaran yang dihasilkan untuk dilakukan perhitungan *error* yang terjadi. Nilai *error* tersebut kemudian masuk ke PID controller untuk selanjutnya aktuator, dalam hal ini servo motor berputar sehingga angin yang diberikan oleh turbin angin dapat diekstrak dan mampu memutar *blade* pada turbin angin. Skema tersebut dijelaskan melalui Blok Diagram sistem kontrol *Pitch* turbin angin yang ditunjukkan oleh Gambar 3.9.



Gambar 3.9 Blok diagram sistem kontrol *Pitch* turbin angin

Plant dalam hal ini merupakan keseluruhan komponen pada sistem turbin angin meliputi aktuator berupa motor servo, *blade* dengan sudut tertentu, serta sensor *rotary encoder* sebagai *transmitter*. Dalam pemodelan ini, keseluruhan *plant* tersebut dimodelkan menjadi sebuah kesatuan *plant*. Kemudian *plant* tersebut dilakukan pengujian dengan *me-monitoring* waktu serta rpm yang dihasilkan hingga mencapai *steady*.

Hasil dari pengujian turbin angin tanpa kontrol sistem, kemudian dimodelkan secara matematis. Pemodelan ini diperlukan untuk mempermudah dalam mendesain kontrol turbin angin nantinya. Pemodelan didapatkan dengan metode FOPDT (*First Order Plus Delay Time*). Untuk mendapatkan model

matematis dari sistem ini, perlu diadakan pengujian respon sistem secara *real* terlebih dahulu.

Pengujian sistem dilakukan pada kecepatan angin 2,6 m/s dengan bukaan aktuator dalam hal ini adalah servo motor sebesar 4^0 . Setelah itu, didapatkan diagram antara waktu dan keluaran sistem berupa rpm mulai *starting* hingga stabil. Kurva tersebut kemudian dianalisa hingga didapatkan model matematisnya dengan persamaan FOPDT berikut ini (Alrijadjis):

$$FOPDT = G_m(s) = K \frac{e^{-\theta s}}{\tau s + 1} \quad (3.1)$$

$$K = \frac{\Delta}{\delta} \quad (3.2)$$

$$\tau = 1,5(t_{63\%} - t_{28\%}) \quad (3.3)$$

$$\theta = t_{63\%} - \tau \quad (3.4)$$

Dimana,

K : *gain steady state*

Δ : perubahan *steady state* pada keluaran proses

δ : perubahan *step* pada masukan

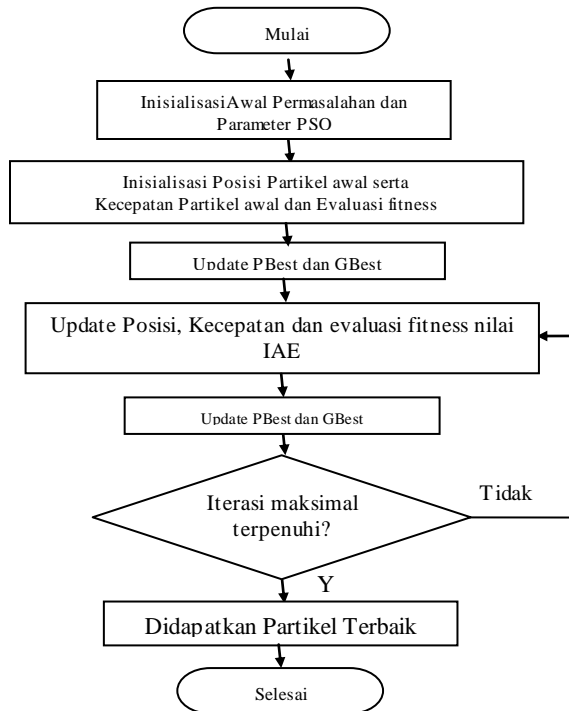
$t_{28\%}$: nilai PV ketika mencapai 28% perubahan awal

$t_{63\%}$: nilai PV ketika mencapai 63% perubahan awal

Dari hasil pemodelan tersebut, selanjutnya dilakukan uji *open loop* sinyal keluaran yang dihasilkan dari pemodelan sistem. Pengujian ini dilakukan menggunakan aplikasi Simulink yang ada pada Matlab. Hasil Uji *open loop* pemodelan tersebut kemudian dibandingkan dengan kurva karakteristik yang didapatkan dari uji *real plant* untuk diketahui apakah hasil pemodelan sudah mewakili *plant* yang sebenarnya. Pengujian hasil *real plant* dilakukan sebanyak 5 kali. Hasil pengujian tersebut kemudian dicari nilai *error* terbesar untuk selanjutnya dianalisa.

3.4 Pencarian Parameter Kontroler PID menggunakan PSO

Algoritma PSO dibangun menggunakan aplikasi Matlab berupa *m file* yang diintegrasikan dengan simulink dari sistem kontrol *pitch* turbin angin. Fungsi Objektif dalam permasalahan ini adalah nilai minimum dari IAE (*Integral Absolute Error*) yang merupakan nilai luasan *error* antara *set point* dengan respon sistem. Sedangkan *variable* yang dioptimasi adalah nilai parameter kontrol PID yakni, nilai gain K_p , T_i dan T_d . Diagram alir *tunning* PID-PSO ditunjukkan oleh Gambar 3.10.



Gambar 3.10 Diagram alir optimalisasi menggunakan PSO

Langkah pertama yang harus dilakukan dalam melakukan *tunning* parameter kontrol PID adalah dengan penentuan jumlah individu dalam satu populasi, penentuan nilai posisi dan kecepatan, penentuan jumlah iterasi dan penentuan nilai bobot inersia. Setelah itu, dilakukan evaluasi nilai *fitness* untuk setiap

nilai *particle* awal yang didapat. Evaluasi nilai *fitness* dalam hal ini merupakan pencarian nilai luasan *error* antara *set point* dengan sinyal yang terbentuk, jadi evaluasi nilai *fitness* tersebut dilakukan menggunakan simulasi pada Simulink. Inilah gunanya integrasi antara *m file* algoritma PSO dengan simulasi simulink dari sistem *close loop* yang dibuat.

Nilai PBest (*Personal Best*) pada inisialisasi awal ini merupakan nilai posisi serta *fitness* masing-masing *particle*. Sedangkan nilai Gbest (*Global Best*) pada inisialisasi awal merupakan nilai posisi terbaik dari Pbest yang didapat sebelumnya.

Kemudian dilakukan *update* nilai kecepatan baru dan posisi baru setiap partikel dengan menggunakan persamaan (2.10) dan (2.11). Hal ini dilakukan hingga iterasi maksimum terpenuhi. Dalam setiap iterasi, dilakukan pula *update Personal Best* serta *Global Best*.

Hasil *update* nilai *Global Best* pada iterasi terakhir merupakan nilai posisi terbaik yang menghasilkan nilai *fitness* paling optimal dari *tunning* parameter kontrol PID menggunakan algoritma PSO tersebut. Sehingga, nilai tersebut yang nantinya digunakan sebagai nilai parameter kontrol PID.

3.5 Realisasi Kontrol Sudut *Pitch* Turbin Angin

Pada tahap ini, dilakukan integrasi antara *hardware* dengan *software*. *Hardware* dalam hal ini terdiri dari *plant* turbin angin, dengan *input* nilai data untuk posisi sudut *blade* turbin angin yang digerakkan oleh motor servo serta output tegangan fotodioda untuk pembacaan nilai rpm, modul arduino, serta sebuah *personal computer* sebagai media monitoring. Program yang diimplementasikan pada modul Arduino berupa program pembacaan nilai rpm, program pengendali PID serta program untuk menampilkan nilai rpm serta waktu yang berlangsung.

Hasil *tuning* parameter kontrol PID menggunakan PSO kemudian diimplementasikan ke dalam prosesor Arduino Uno. Implementasi tersebut dengan menggunakan bahasa pemrograman yang dapat diterjemahkan oleh prosesor yakni bahasa pemrograman C.

Setpoint pada tahap ini dimasukkan besaran nilai rpm berapa yang diinginkan untuk dijaga tetap oleh sistem turbin angin. Kemudian dibandingkan dengan rpm yang sedang dihasilkan yang kemudian dicari nilai selisihnya, atau dalam hal ini disebut nilai *error*. Nilai *error* tersebut kemudian dimasukkan ke dalam Kontrol PID untuk diubah sinyalnya menjadi nilai masukan bagi motor servo.

Dalam tahap ini, logika kontroler PID terlebih dahulu diubah menjadi bahasa C. Sehingga, diperlukan terlebih dahulu persamaan kontroler PID dalam domain waktu diskrit. Berikut ini persamaannya (Sunarto, 2011):

$$U(k) = K_p e(k) + K_i T_c [e(0) + \dots + e(k-1) + e(k)] + K_d [e(k) - e(k-1)] / T_c \quad (3.5)$$

Dimana,

- $U(k)$: Sinyal diskrit
- K_p : *Gain* kontrol proporsional
- K_i : *Gain* kontrol integral
- K_d : *Gain* kontrol *derivative*
- $e(k)$: *error* iterasi saat ini
- $e(k-1)$: *error* iterasi sebelumnya

Persamaan tersebut kemudian diubah kedalam bahasa pemrograman C untuk selanjutnya dijadikan *input* aktuator dalam hal ini adalah nilai posisi motor servo.

Selanjutnya, dilakukan pengujian sistem kontrol sudut *pitch* secara *hardware* dan *software*. Pengujian yang dilakukan dengan memberikan nilai target atau *set point* pada program arduino, kemudian dilakukan *monitoring* nilai rpm dari sistem setiap detik. Hasil monitoring nilai rpm tersebut selanjutnya digambarkan dalam bentuk diagram.

Hasil pengujian dianalisa kriteria performansinya yang kemudian dibandingkan dengan hasil pengujian saat sebelum diberikan kontrol. Pengujian tersebut berupa indeks performansi seperti *time settling*, *maximum overshoot*, serta nilai *rise time*-nya.

BAB IV HASIL DAN PEMBAHASAN

4.1 Hasil Pengujian Perancangan *Hardware*

4.1.1 Hasil Pengujian *Rotary Encoder*

Pengujian *rotary encoder* ini menggunakan sebuah motor listrik arus bolak-balik. Motor tersebut diberi tegangan bervariasi sehingga menghasilkan perputaran yang bervariasi pula. Pengukuran rpm (*Rotation Per Minute*) dilakukan oleh dua alat ukur yakni menggunakan sensor *rotary encoder* dan juga menggunakan Tachometer. Pengambilan data dilakukan 3 kali pada setiap variasi kecepatan yang kemudian dirata-rata. Berikut ini hasil pengujian sensor *rotary encoder* yang ditunjukkan oleh Tabel 4.1.

Tabel 4.1 Hasil Pengujian Sensor *Rotary Encoder*

Pengujian ke-	Daya motor	rpm	
		Rotary Encoder	Tachometer
1	I = 0,3 A V = 1,7 V	390	393
2	I = 0,4 A V = 2,5 V	570	574
3	I = 0,35 A V = 3,7 V	1048	1055
4	I = 0,39 A V = 8,0 V	2644	2658
5	I = 0,39 A V = 6,0 V	1870	1837

Dari hasil pengujian di atas, nilai rpm hasil keluaran sensor *rotary encoder* yang telah dirancang menunjukkan nilai yang tidak jauh berbeda dengan nilai yang ditunjukkan oleh alat ukur Tachometer. Sehingga, sensor *rotary encoder* dianggap telah mampu melakukan pembacaan nilai rpm turbin angin secara akurat.

4.1.2 Pengujian Sudut *Pitch* Optimum Turbin Angin

Pengujian ini dilakukan dengan memberikan sumber angin pada *plant* turbin angin. *Plant* turbin angin juga diintegrasikan dengan Arduino yang telah diberi program pembacaan nilai rpm serta program pengatur posisi sudut servo motor. Pada pengujian ini, terdapat 3 variasi sumber kecepatan angin yang diberikan. Turbin angin dikondisikan dengan perputaran blade setiap kenaikan $1,4^0$. Hasil implementasi ini adalah nilai sudut *pitch blade* turbin angin paling optimal yang mampu menghasilkan putaran turbin angin optimal. Berikut ini hasil pengujian sudut maksimum yang ditunjukkan oleh Tabel 4.2.

Tabel 4.2 Pengujian Sudut *Pitch* Maksimum Sistem Turbin Angin

Posisi Sudut (0)		Rpm					
Servo	Blade	V1 = 2.6 m/s	V1 = 2.3 m/s	V1 = 1.9 m/s	V1 = 1.8 m/s	V1 = 1.7 m/s	V1 = 1.6 m/s
0	0	15	10	12	17	11	12
2	1.4	25	18	22	16	14	20
4	2.8	86	87	86	88	91	83
6	4.2	94	92	84	71	84	78
8	5.6	79	82	77	79	78	71
10	7	68	72	74	73	71	63
12	8.4	58	60	58	58	60	60
14	9.8	55	56	54	53	54	51
16	11.2	50	51	57	48	50	47
18	12.6	44	47	55	45	46	44
20	14	43	44	54	43	42	39
22	15.4	39	39	40	39	38	34
24	16.8	37	37	39	37	36	32
26	18.2	35	34	36	35	35	31
28	19.6	34	31	32	32	32	29

Tabel 4.2 Pengujian Sudut *Pitch* Maksimum Sistem Turbin Angin (Lanjutan)

Posisi Sudut (°)		rpm					
Servo	Blade	V1 = 2.6 m/s	V1 = 2.3 m/s	V1 = 1.9 m/s	V1 = 1.8 m/s	V1 = 1.7 m/s	V1 = 1.6 m/s
30	21	31	30	29	31	31	24
32	22.4	32	29	29	29	29	22
34	23.8	29	28	27	28	29	22
36	25.2	28	27	27	27	27	21
38	26.6	27	25	26	25	23	20
40	28	25	25	24	24	24	19
42	29.4	25	24	23	21	21	17
44	30.8	24	21	22	21	21	17
46	32.2	22	20	21	20	19	19
48	33.6	21	19	19	20	20	19
50	35	21	18	18	19	17	17
52	36.4	21	19	20	18	17	17
54	37.8	20	17	18	18	19	13
56	39.2	19	17	18	18	17	15
58	40.6	19	16	17	17	16	16
60	42	18	16	16	16	16	15
62	43.4	17	15	14	15	16	16
64	44.8	17	15	15	14	13	13
66	46.2	16	14	15	14	14	14
68	47.6	14	15	13	13	12	11
70	49	14	14	13	13	12	11
72	50.4	13	13	13	11	11	10
74	51.8	12	13	13	11	11	10
76	53.2	10	7	10	10	9	0
78	54.6	9	8	8	12	8	0
80	56	2	2	0	8	0	0

Hasil pengujian menunjukkan bahwa nilai sudut maksimum dimana turbin angin mampu menghasilkan putaran maksimum yakni pada sudut *blade* $4,2^0$ untuk sumber kecepatan angin 2,3 hingga 2,6 m/s dan juga pada sudut *blade* $2,8^0$ untuk sumber kecepatan angin 1,6 m/s hingga 2,3 m/s. Hal tersebut membuktikan bahwa kecepatan putaran turbin angin memiliki sudut *pitch* optimum yang berbeda-beda untuk setiap sumber kecepatan angin yang berbeda-beda pula.

4.1.3 Pengujian Nilai *Power Coefficient* (Cp)

Pada pengujian ini, dilakukan pengukuran kecepatan angin sebelum dan sesudah melewati turbin angin. Pengambilan data kecepatan angin dilakukan menggunakan alat ukur kecepatan angin berupa anemometer serta dilakukan pada saat turbin angin dalam kondisi berputar dengan putaran terbesarnya. Perhitungan nilai Cp dilakukan sesuai dengan persamaan (2.4). Berikut ini tiga contoh uraian perhitungannya:

- Saat $V_1 = 2.6 \text{ m/s}$ dan $V_2 = 1.2 \text{ m/s}$

$$\begin{aligned} CP &= 0.5 (1+v_2/v_1) (1-(v_2/v_1)^2) \\ &= 0.5 (1+1.2/2.6) (1-(1.2/2.6)^2) \\ &= \mathbf{0.57510} \end{aligned}$$
- Saat $V_1 = 1.9 \text{ m/s}$ dan $V_2 = 0.9 \text{ m/s}$

$$\begin{aligned} CP &= 0.5 (1+v_2/v_1) (1-(v_2/v_1)^2) \\ &= 0.5 (1+0.9/1.9) (1-(0.9/1.9)^2) \\ &= \mathbf{0.57151} \end{aligned}$$
- Saat $V_1 = 1.6 \text{ m/s}$ dan $V_2 = 0.8 \text{ m/s}$

$$\begin{aligned} CP &= 0.5 (1+v_2/v_1) (1-(v_2/v_1)^2) \\ &= 0.5 (1+0.8/1.6) (1-(0.8/1.6)^2) \\ &= \mathbf{0.56250} \end{aligned}$$

Hasil pengujian kecepatan angin sebelum dan sesudah turbin angin serta perhitungan Cp – nya ditunjukkan oleh Tabel 4.3.

Tabel 4.3 Hasil Pengujian Nilai Cp Turbin Angin

v_1 (m/s)	2.6	2.3	1.9	1.8	1.7	1.6
v_3 (m/s)	1.2	1.3	0.9	1.0	0.9	0.8
CP	0.57510	0.53289	0.57151	0.53772	0.55038	0.56250

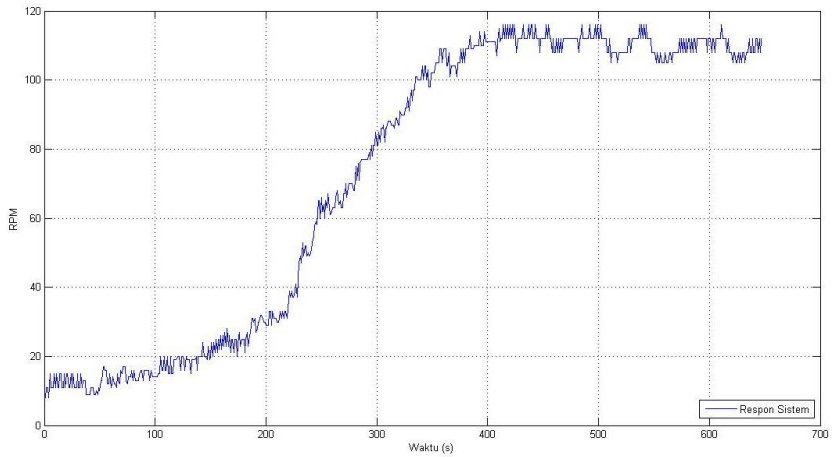
Jika dibandingkan dengan nilai Cp maksimum (*Betz Limit*) yaitu sebesar 0.592592592592593, maka nilai Cp dari turbin angin ini mendekati nilai Cp maksimum tersebut. Sehingga dari pengujian disimpulkan bahwa turbin angin memiliki performa yang sangat baik karena mampu menghasilkan nilai Cp mendekati nilai Cp maksimum.

4.1.4 Pengujian *Prototype* Turbin Angin

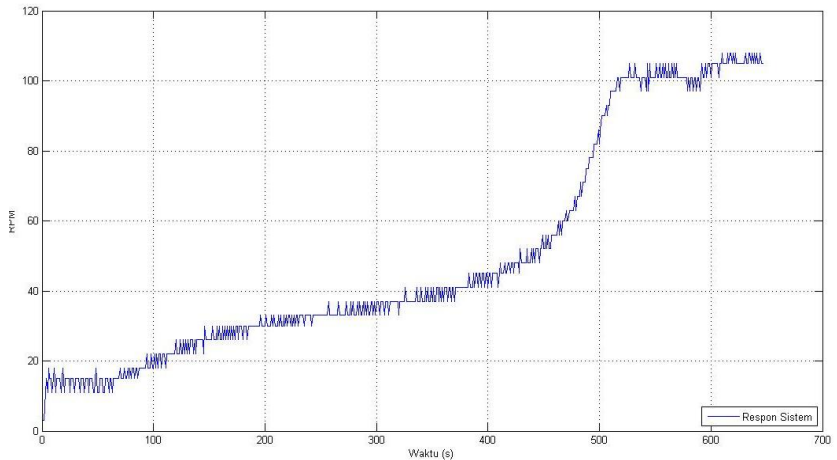
Hasil pengujian sudut *pitch* optimal turbin angin pada tiga jenis kecepatan angin yakni pada kecepatan 2,6 m/s; 1,9 m/s dan 1,6 m/s. Untuk mengetahui respon sistem secara keseluruhan pada daerah optimal kinerja turbin angin, maka dilakukan pengujian respon sistem turbin angin terhadap waktu dengan *input step* sudut servo 0° ke 4° .

Pengujian dilakukan dengan mengintegrasikan *plant* turbin angin dengan Arduino. Perangkat Arduino diberikan program untuk memonitoring nilai rpm di setiap waktu serta diberikan program untuk mengatur sudut motor servo sebesar 4° . Hasil monitoring dibaca melalui perangkat *Personal Computer* yang terhubung dengan layar serial aplikasi Arduino. *Input* dari sistem pengujian ini adalah posisi sudut *pitch* turbin angin serta kecepatan angin. Sedangkan *output* dari pengujian ini adalah nilai rpm sistem di setiap waktu.

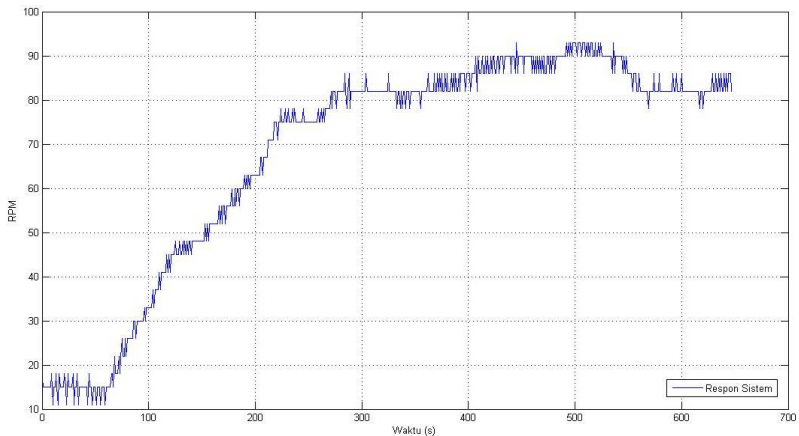
Pengujian dilakukan dalam kurun waktu 650 detik yakni hingga respon *steady*. Hasil data *monitoring* kemudian di-*plot* menggunakan aplikasi matlab. Hasil respon sistem menunjukkan respon yang berbeda-beda sesuai dengan kurva yang ditunjukkan secara berturut-turut oleh Gambar 4.1, Gambar 4.2 dan Gambar 4.3.



Gambar 4.1 Pengujian sistem turbin angin dengan perubahan sudut 4^0 pada kecepatan angin 2,6 m/s



Gambar 4.2 Pengujian sistem turbin angin dengan perubahan sudut 4^0 pada kecepatan angin 1,9 m/s



Gambar 4.3 Pengujian sistem turbin angin dengan perubahan sudut 4° pada kecepatan angin 1,6 m/s

Dari hasil kurva tersebut, diketahui sistem memiliki performansi yang paling baik pada saat diberikan sumber dengan kecepatan angin sebesar 2,6 m/s. Pada saat diberikan sumber angin dengan kecepatan di bawah kecepatan tersebut, sistem tidak menunjukkan respon yang *steady*. Hal tersebut terlihat pada saat sistem diberikan sumber dengan kecepatan angin 1,6 m/s. Sedangkan respon sistem dengan kecepatan angin sebesar 1,9 m/s, menunjukkan bahwa sistem mampu mencapai titik *steady* akan tetapi memerlukan waktu yang lama dibandingkan dengan sistem dengan sumber kecepatan angin sebesar 2,6 m/s.

4.2 Perancangan Sistem Kontrol *Pitch* Turbin Angin

4.2.1 Pemodelan FOPDT

Pemodelan dilakukan menggunakan metode FOPDT dengan acuan respon sistem pada saat turbin angin diberikan sumber kecepatan angin sebesar 2,6 m/s yang ditunjukkan oleh Gambar 4.1. Pemodelan *plant* turbin angin ini mencakup seluruh komponen turbin angin meliputi motor servo, blade turbin angin yang terpasang serta sensor *rotary encoder*. Hasil Pemodelan

merupakan respon sistem orde satu dengan *gain* dan *time constan* mendekati respon *real plant*. Berikut ini uraian pemodelan FOPDT pada turbin angin:

Persamaan umum pemodelan FOPDT berdasarkan persamaan (3.1) adalah:

$$FOPDT = G_m(s) = K \frac{e^{-\theta s}}{\tau s + 1}$$

Berdasarkan persamaan (3.2), diketahui bahwa nilai *gain* merupakan pembagian antara *span output* terhadap *span input*.

$$K = \frac{\Delta}{\delta}$$

Jika dilihat dari Gambar 4.1, diketahui bahwa *span output* bernilai merupakan selisih *output* sinyal saat *steady* untuk *input* 4⁰ dengan selisih sinyal saat *steady* untuk *input* 0⁰. Kemudian dibagi *span* / selisih *input*-nya. Nilai selisih *span output* sebesar 110 rpm dikurangi 10 rpm. Sedangkan selisih *span input* sebesar 4⁰ dikurangi 0⁰. Sehingga,

$$K = \frac{\Delta}{\delta} = (110 - 10) / (4 - 0) = 25$$

Untuk perhitungan *time constan*, berikut ini persamaannya:

$$\tau = 1,5(t_{63\%} - t_{28\%})$$

Nilai rpm pada saat mencapai 63% *steady* yakni sebesar 73 sedangkan nilai rpm pada saat mencapai 28% *steady* yakni sebesar 38 (Data dan Penjelasan terdapat di LAMPIRAN C). Sehingga, $t_{63\%}$ dan $t_{28\%}$ didapatkan dari hasil interpolasi sinyal respon sistem terhadap waktu. Berikut tabel interpolasi yang ditunjukkan berturut-turut Tabel 4.4 dan Tabel 4.5.

Tabel 4.4 Interpolasi Waktu untuk Respon 63% *Steady*

T	rpm
282	71
t = 282,4	73
283	76

Tabel 4.5 Interpolasi Waktu untuk Respon 28% *Steady*

T	Rpm
220	34
t = 220,8	38
221	39

Dari perhitungan, diperoleh nilai $t_{63\%}$ sebesar 282,4 s sedangkan $t_{28\%}$ sebesar 220,8 s. Sehingga, nilai *time constante* dari persamaan (3.3) untuk sistem ini adalah:

$$\tau = 1,5(t_{63\%} - t_{28\%}) = 1,5 (282,4 - 220,8) \\ = 92,4$$

Untuk menghitung nilai *time delay* pada pemodelan plant ini, berikut ini persamaan umum serta perhitungannya:

$$\theta = t_{63\%} - \tau = 282,4 - 92,4 = 190$$

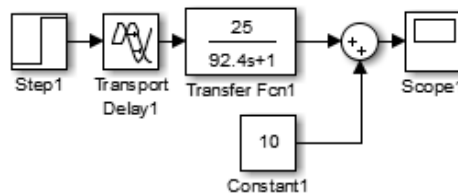
Jadi, model matematis secara keseluruhan sistem turbin angin adalah:

$$FOPDT = G_m(s) = K \frac{e^{-\theta s}}{\tau s + 1} \\ G_m(s) = 25 e^{-190 \text{ s}} / (92,4 \text{ s} + 1)$$

4.2.2 Uji *Open Loop* Pemodelan FOPDT

Pengujian ini bertujuan untuk mengetahui apakah pemodelan tersebut mewakili *plant* secara *real* atau belum. Pengujian dilakukan dengan membandingkan respon sistem hasil simulasi pemodelan menggunakan simulink yang diambil sebanyak sekali dengan respon hasil *real plant* yang diambil sebanyak 5 kali. Keseluruhan respon sistem tersebut selanjutnya dibandingkan dan dicari nilai *error* maksimum beserta *error* rata-rata.

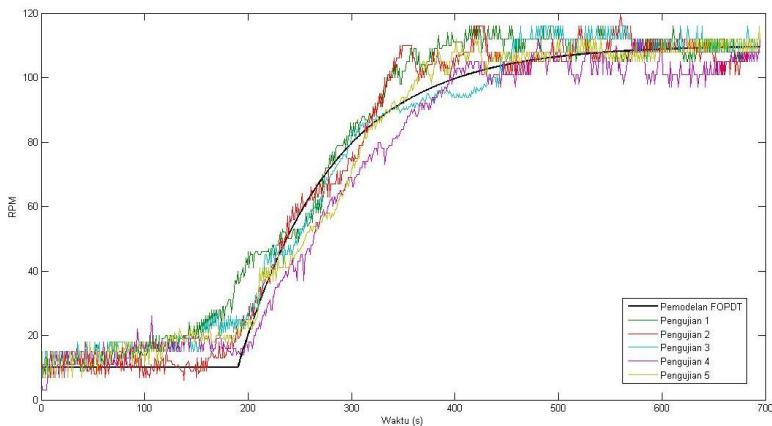
Hasil uji *open loop* pemodelan sistem menggunakan simulink pada matlab yang ditunjukkan oleh Gambar 4.4. Pada blok diagram tersebut merepresentasikan pemodelan *real plant* dengan *input* sudut *pitch* sebesar 4^0 untuk sistem dengan kecepatan angin sebesar 2,6 m/s.



Gambar 4.4 Simulink uji sistem pemodelan secara *open loop*

Sinyal *step* pada blok diagram tersebut meruakan *input* motor servo sebesar 4^0 . Sedangkan *Transport Delay 1* menunjukkan nilai waktu tunda dari hasil pemodelan sistem. Fungsi transfer diberikan sesuai dengan nilai *gain* dan *time constan* pada hasil pemodelan. Kemudian *input* konstanta 10 menunjukkan bahwa sistem pada saat mendapat *input* sudut *pitch* minimum 0^0 telah mampu menghasilkan output rpm sebesar 10.

Simulasi kemudian dijalankan sehingga diperoleh data nilai respon sistem setiap waktunya. Selanjutnya, hasil pemodelan tersebut di-plot dengan hasil pengujian *real plant* turbin angin yang dilakukan sebanyak 5 kali. Kurva hasil pengujian untuk *input* motor servo 4^0 , ditunjukkan oleh Gambar 4.5.



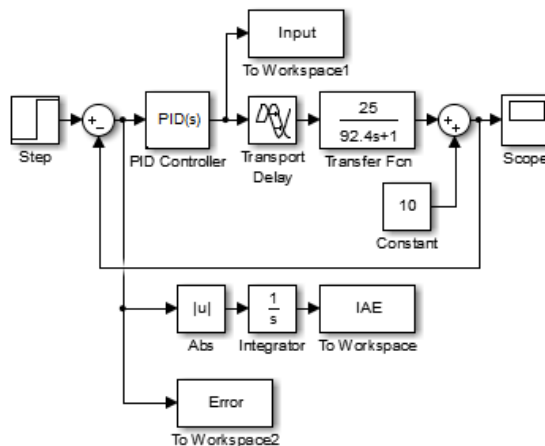
Gambar 4.5 Kurva hasil uji *open loop* sistem turbin angin *input* sudut pitch 4^0

Hasil pengujian *real plant* dengan hasil pemodelan dibandingkan dengan cara menghitung selisih atau *error* rpm yang terjadi di beberapa titik *sampling* waktu. Pada *sampling* waktu tersebut didapatkan nilai *error* maksimum.

Tabel 4.6 Error Pengujian Input 4⁰

Waktu (s)	rpm Pengujian Ke-					rpm Pemodelan	Error Maks
	1	2	3	4	5		
50 s	10	12	11	11	11	10	20.00%
270 s	58	67	63	57	56	68	17.55%
450 s	112	101	105	101	101	104	7.69%
550 s	108	108	108	112	101	108	6.45%
690 s	112	108	112	108	112	110	2.24%

Perhitungan *error* menunjukkan bahwa pada awal waktu (50 detik), *error* maksimal dari lima pengujian sebesar 20%. Akan tetapi, *error* tersebut semakin menurun seiring berjalannya waktu. Dari hasil perbandingan tersebut, pemodelan telah dianggap mewakili *real plant*.



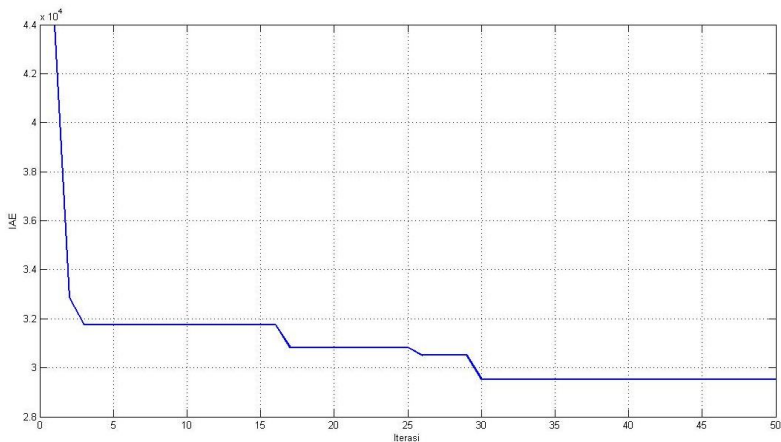
Gambar 4.6 Diagram simulink sistem *close loop* turbin angin

Hasil pemodelan *open loop* kemudian dijadikan pemodelan *close loop* dengan pengendali PID. Blok diagram pemodelan

sistem secara *close loop* ditunjukkan oleh Gambar 4.6. Pada blok diagram tersebut, pemodelan *plant* akan sama seperti pemodelan secara *open loop* akan tetapi, sinyal keluran sistem diumpun balikkan ke *input* sehingga *input* sistem merupakan nilai *error* antara *set point* dengan sinyal rpm yang dihasilkan.

4.3 Tuning PID Menggunakan Metode PSO

Tuning PID-PSO dilakukan dengan membuat sebuah *m file* yang berisi algoritma optimasi PSO kemudian diintegrasikan dengan simulink yang berisikan pemodelan *close loop system* turbin angin pada aplikasi matlab. Fungsi objektif untuk menentukan *fitness* dari optimalisasi PSO ini merupakan nilai IAE yang didapat dari hasil simulasi simulink. Variabel yang dioptimasi yakni parameter kontroller PID yaitu nilai K_p , K_i dan K_d . Batasan-batasan nilai variabel didapatkan dari hasil *tunning* secara *trial and error*.



Gambar 4.7 Grafik penurunan nilai IAE terhadap jumlah Iterasi

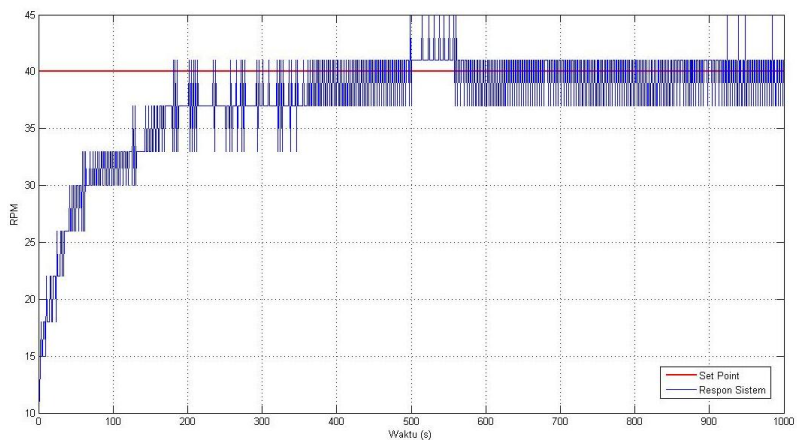
Program matlab dari algoritma PSO ditunjukkan di lampiran. Optimalisasi dilakukan sebanyak 50 iterasi. Hasil dari optimalisasi tersebut didapatkan penurunan nilai IAE dari iterasi

awal hingga terakhir. Grafik penurunan IAE yang ditunjukkan oleh Gambar 4.7.

Nilai IAE pada saat pertama kali dilakukan optimasi, sebesar 43979 dan mengalami penurunan hingga mencapai nilai IAE sebesar 29525. Nilai IAE tersebut didapatkan dengan memberikan parameter PID berupa nilai K_p sebesar 0,0239069; nilai K_i sebesar 0,0001382 serta nilai K_d sebesar 0,053518. Nilai Parameter PID tersebut selanjutnya diterapkan ke sistem turbin angin secara *real*.

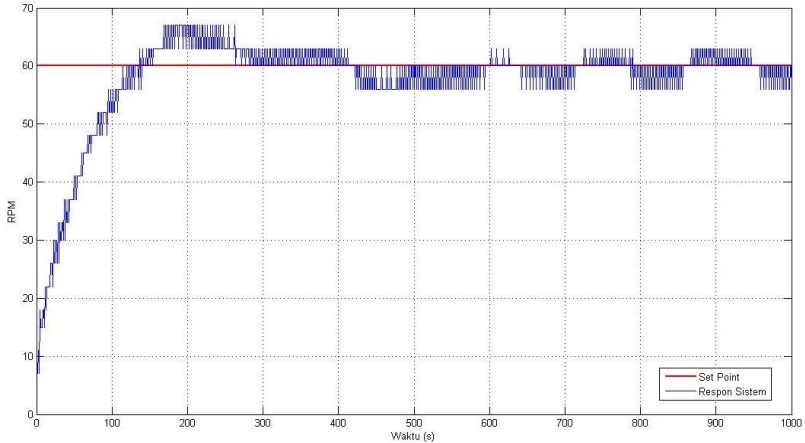
4.4 Pengujian Tracking Set Point Sistem Kontrol Turbin Angin

Pengujian awal dilakukan pada sistem turbin angin dengan memberikan nilai *set point* sebesar 40 rpm, 60 rpm, 80 rpm serta 100 rpm. Logika kontrol PID diterapkan pada modul Arduino yang terhubung secara langsung dengan *plant* turbin angin yang digerakkan oleh sumber angin dengan kecepatan sebesar 2,6 m/s. Pengambilan data dilakukan selama 1000 detik. Hasil *monitoring* selanjutnya di-plot menggunakan aplikasi matlab. Hasil pengujian respon *Set Point* 40 rpm ditunjukkan oleh Gambar 4.8.

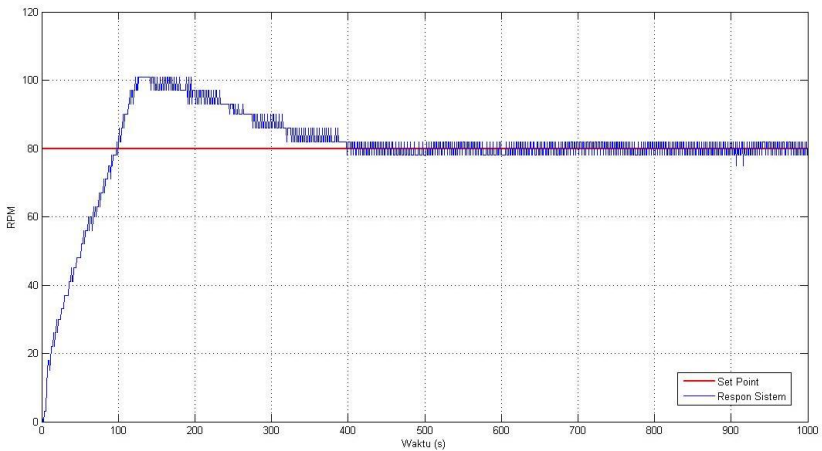


Gambar 4.8 Pengujian *set point* 40 rpm

Dari grafik tersebut, didapatkan bahwa sistem mampu mencapai *set point* dengan *overshoot* maksimal 12.5%. Sistem juga mulai konstan mencapai nilai *set point* pada waktu ke 181 detik. Sedangkan grafik hasil pengujian tracking *set point* 60 rpm ditunjukkan oleh Gambar 4.9.



Gambar 4.9 Pengujian *set point* 60 rpm

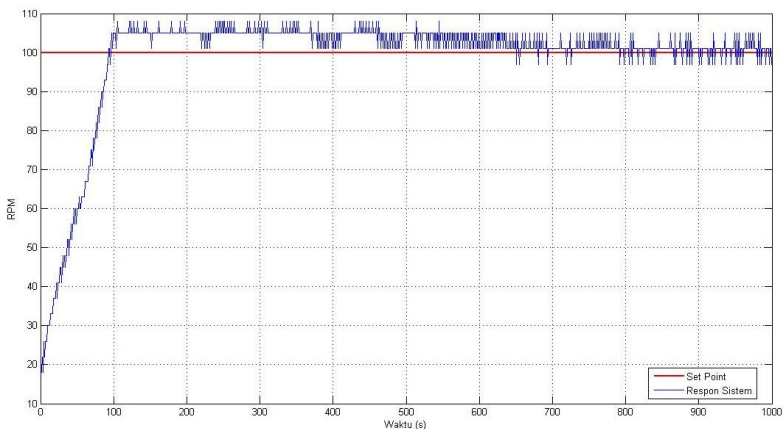


Gambar 4.10 Pengujian *set point* 80 rpm

Hasil pengujian *set point* 60, menunjukkan bahwa respon sistem mampu mencapai nilai *set point* hingga detik ke-114 detik dengan *overshoot* maksimal sebesar 10%.

Hasil pengujian respon sistem dengan *set point* 80 rpm ditunjukkan oleh Gambar 4.10. Dari grafik hasil tersebut, diketahui bahwa respon sistem mampu mencapai nilai *set point* hingga detik ke-98 detik dengan *overshoot* mencapai 20%.

Hasil pengujian *set point* selanjutnya yakni *set point* 100 rpm. Grafik hasil pengujian tersebut ditunjukkan oleh Gambar 4.11. Dari grafik hasil pengujian tersebut, diketahui bahwa respon sistem mampu mencapai nilai *set point* hingga detik ke-93 detik dengan *overshoot* maksimum sebesar 10%.



Gambar 4.11 Pengujian *set point* 100 rpm

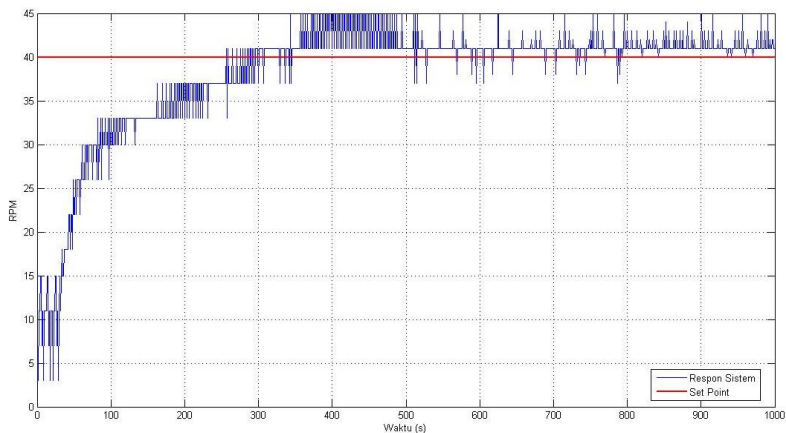
Hasil pengujian setiap nilai *set point* menunjukkan respon sistem yang berbeda-beda. Untuk mempermudah dalam menganalisa, maka diperlukan perbandingan kriteria indeks performansi di setiap pengujian. Indeks performansi pada pengujian ini meliputi *rise time*, *maksimum overshoot* serta *settling time*. *Rise time* merupakan waktu yang dibutuhkan oleh sistem untuk pertama kali mencapai nilai *set point*. Sedangkan *maksimum overshoot* merupakan kenaikan maksimum respon sistem yang melebihi nilai *set point*. Sedangkan *settling time*

merupakan waktu yang dibutuhkan suatu sistem untuk mencapai *steady state*. Perbandingan indeks performansi untuk pengujian setiap nilai *set point* yang merupakan respon dari Gambar 4.8, Gambar 4.9, Gambar 4.10 dan Gambar 4.11 ditunjukkan oleh Tabel 4.7.

Tabel 4.7 Indeks Performansi Hasil Pengujian *Set Point*

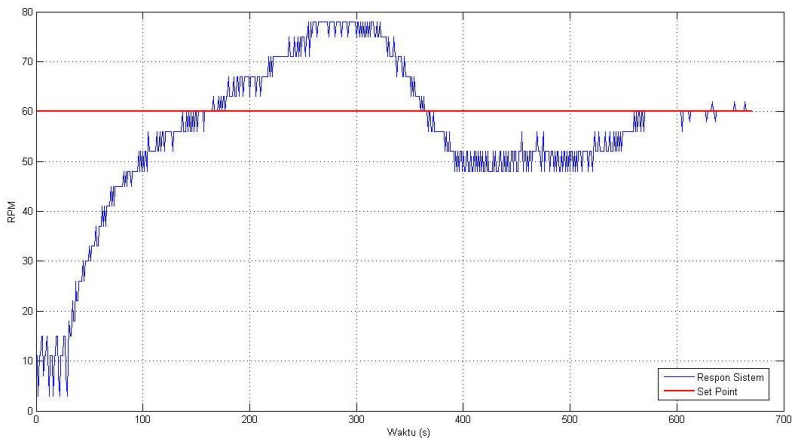
<i>Set Point</i> (rpm)	<i>Rise Time, t_r</i> (s)	<i>Maksimum Overshoot, M_p</i> (%)	<i>Settling Time, t_s</i> (s)
40	181	12,5	347
60	114	11,7	264
80	98	26,25	387
100	93	8,00	93

Hasil pengujian menunjukkan bahwa sistem mampu merespon nilai *set point* lebih baik saat diberi kontrol PID. Hal tersebut ditunjukkan oleh parameter *rise time* yang jauh lebih cepat dibanding sistem *open loop* yang memiliki *rise time* sebesar 400 detik. Maksimum *Overshoot* terbesar terjadi pada saat sistem diberikan nilai *set point* 80 rpm yakni sebesar 25,25%.

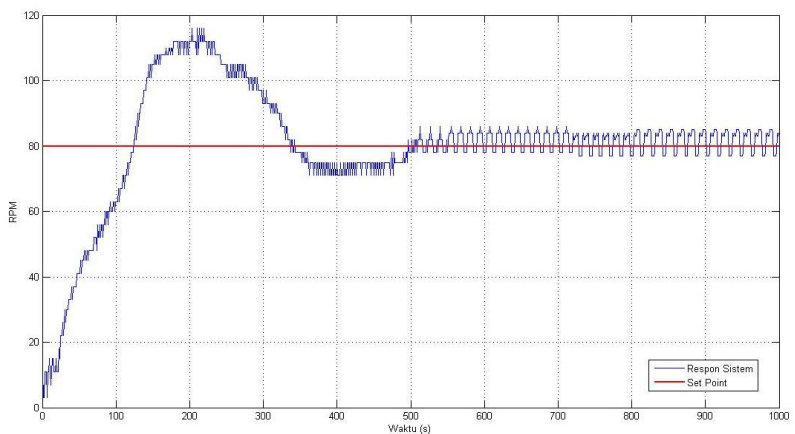


Gambar 4.12 Pengujian *set point* 40 rpm metode PI-Fuzzy

Percobaan kontrol pitch turbin angin juga dilakukan dengan melakukan tuning PI menggunakan metode Fuzzy. Uji respon terhadap nilai *set point* yang diberikan juga dilakukan untuk metode fuzzy. Hasil uji diperlihatkan oleh Gambar 4.12 untuk *set point* 40 rpm, Gambar 4.13 untuk *set point* 60 rpm dan Gambar 4.14 untuk *set point* 80 rpm.



Gambar 4.13 Pengujian *set point* 60 rpm metode PI-Fuzzy



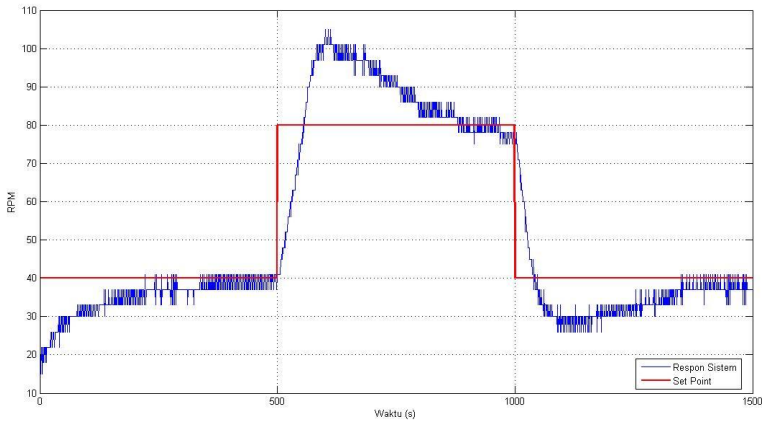
Gambar 4.14 Pengujian *set point* 80 rpm metode PI-Fuzzy

Hasil dari respon yang didapat dengan metode PI-Fuzzy yang ditunjukkan oleh Gambar 4.12, Gambar 4.13 dan Gambar 4.14 selanjutnya dapat dibandingkan dengan respon yang didapat dari metode PID-PSO yang ditunjukkan oleh Gambar 4.8, Gambar 4.9 dan Gambar 4.10 dan. Hasil perbandingan tersebut ditunjukkan oleh Tabel 4.8.

Tabel 4.8 Perbandingan Kriteria Performansi Hasil Tuning Parameter PID Menggunakan PSO dan Fuzzy

<i>Set Point</i> (rpm)	<i>Rise Time, t_r</i> (s)		<i>Maksimum</i> <i>Overshoot, M_p</i> (%)		<i>Settling Time, t_s</i> (s)	
	PSO	Fuzzy	PSO	Fuzzy	PSO	Fuzzy
40	181	256	12,5	12,5	347	345
60	114	149	11,7	30	264	549
80	98	107,5	26,25	42	387	323

Hasil perbandingan yang ditunjukkan oleh Tabel 4.8 menunjukkan bahwa hasil tuning parameter PID menggunakan PSO akan menghasilkan respon yang lebih optimal jika dibandingkan dengan menggunakan fuzzy. Parameter yang dijadikan perbandingan yakni nilai *rise time*, *maksimum overshoot* serta *settling time* yang terjadi.



Gambar 4.15 *Tracking set point 40-80-40 rpm*

Pengujian kontrol sudut *pitch* turbin angin juga dilakukan dengan melakukan uji *tracking set point*. Respon sistem uji *tracking set point* ditunjukkan oleh Gambar 4.15.

Dari hasil uji *tracking set point* tersebut, diketahui bahwa sistem mampu mencapai nilai *set point* yang diinginkan, dengan lama waktu ke 269 sesaat setelah diberikan *set point* awal, kemudian sistem turbin angin akan mampu mencapai *set point* kedua dalam rentang waktu 55 detik. Saat *set point* diturunkan lagi, diketahui sistem turbin angin mampu mencapai *set point* pada waktu ke-39 detik setelah *set point* diberikan. Maksimum *overshoot* yang terjadi pada pengujian ini adalah 31,25% yakni pada saat sistem menuju *set point* 80 rpm.

Halaman ini memang dikosongkan

BAB V

PENUTUP

5.1 Kesimpulan

Dari hasil perobaan yang dilakukan pada penelitian ini, didapatkan kesimpulan sebagai berikut ini:

- a. Telah dirancang sebuah turbin angin dengan sumber angin kecepatan rendah yang telah diuji menggunakan enam variasi kecepatan angin yakni 2,6; 2,3; 1,9; 1,8; 1,7 serta 1,6 m/s dengan nilai sudut *pitch* optimal yang mampu menghasilkan nilai rpm pada masing-masing kecepatan angin, yang ditunjukkan oleh Tabel 4.2, dimana pada tabel tersebut disebutkan bahwa sudut *pitch* optimal sistem turbin angin dengan kecepatan antara 1,6 m/s hingga 2,6 m/s adalah $2,8^0$ hingga $4,2^0$.
- b. Telah dirancang kontrol sudut *pitch* turbin angin skala kecil menggunakan optimisasi parameter PID dengan metode PSO yang menghasilkan nilai Kp 0,0239069; nilai Ki 0,0001382 dan nilai Kd 0,053518. Setelah dilakukan uji *set point* 40, 60, 80 serta 100 rpm, didapatkan performansi sistem yang ditunjukkan oleh Tabel 4.7, dimana sistem menghasilkan nilai *maximum overshoot* terbesar dan *settling time* terlama pada saat *set point* 80 rpm, sedangkan *rise time* terlama ditunjukkan pada saat sistem diuji dengan *set point* 40 rpm.

5.2 Saran

Saran yang dapat diterapkan untuk penelitian selanjutnya adalah:

- a. Dilakukan perbandingan antara respon hasil simulasi pemodelan serta respon dari *real plant*.
- b. Dilakukan uji sistem turbin angin untuk sumber kecepatan angin yang lebih besar.
- c. Dilakukan aplikasi kinerja turbin angin yang diintegrasikan dengan turbin generator listrik sehingga mampu menghasilkan daya listrik.

Halaman ini memang dikosongkan

DAFTAR PUSTAKA

- Musyafa', Ali. R.M, Farid. Negara, I.M.Yulistya. Robandi, Imam. (2011). *Traceability of pitch angle position for small scale wind turbine to geta maximum Energy Extraction*. Deutscher Akademischer Austausch Dienst German Academic Exchange Service.
- S. V. Devika, Sk. Khamuruddeen, Sk. Khamurunnisa, Jayanth Thota, Khalesha Shaik. (2014). *Arduino Based Automatic Plant Watering System. International Journal of Advanced Research in Computer Science and Software Engineering. Volume 4 Issue 10, October 2014*.
- Alrijadjis. Astrowulan, K. Optimasi Kontroler PID Berbasis *Particle Swarm Optimization* (PSO) untuk Sistem dengan Waktu Tunda. Surabaya: Institut Teknologi Sepuluh Nopember.
- Sigit, Rianto. dkk. (2007). *Motor Servo*. Surabaya: Politeknik Elektronika Negeri Surabaya.
- Sunarto. Musyafa', Ali. (2014). Rancang Bangun Sistem Pengendalian Sudut *Pitch* Turbin Angin *Horizontal Axis* Berbasis *Particle Swarm Optimization* (PSO).
- Susanto, Erwin. Kontrol Proporsional Integral Derivatif (PID) Untuk Motor DC Menggunakan *Personal Computer*. Bandung: Institut Teknologi Telkom Bandung.
- da Rosa, Aldo Vieira. 2009. *Fundamentals Of Renewable Energy Process (second edition)*. USA: Stanford University.
- Musyafa', Ali. (2016). *Turbin Angin & Sistem Kontrol*. Surabaya: PT Revka Petra Media.
- Andrade, T F. (2016). *Encoder Working Principle Theory*: https://remotelab.fe.up.pt/remote_exp/Encoder_Working_Principle_Theory.pdf
- Djuandi, Feri. (2011, Juli). *Pengenalan Arduino*: <http://tobuku.com/docs/Arduino-Pengenalan.pdf>
- Karyatulisilmiah. (2015, Oktober). *Particle Swarm Optimization*: <http://karyatulisilmiah.com/particle-swarm-optimization/>

Powerbyproxi. (2016). Slip Ring: <https://powerbyproxi.com/slip-ring/>

LAMPIRAN A Program *Tunnig* PID - PSO

```
clc;
clear;
close all;

%% PROBLEM DEFINITION
nVar = 3;
VarSize = [1 nVar];
kp_max = 0.051121287;
kp_min = 0;
ki_max = 0.000177232;
ki_min = 6.3754E-05;
kd_max = 1.361090329;
kd_min = -5.193793905;

%% PARAMETER
maxIt = 50;
nPop = 20;
w = 0.7;
wdamp = 0.99;
C1 = 2;
C2 = 2;

%% INITIALIZATION
empty_particle.Position = [];
empty_particle.Velocity = [];
empty_particle.Cost = [];
empty_particle.Best.Position = [];
empty_particle.Best.Cost = [];

GlobalBest.Cost = inf;
```

```

%Create Population Array
particle = repmat(empty_particle, nPop, 1);

%Initialize Population Members

for i=1:nPop
    %Generate Random Solution
    particle(i).Position(1) = unifrnd(kp_min, kp_max);
    particle(i).Position(2) = unifrnd(ki_min, ki_max);
    particle(i).Position(3) = unifrnd(kd_min, kd_max);
    %Initialize Velocity
    particle(i).Velocity = zeros(VarSize);

    %Evaluation
    %particle(i).Cost = CostFunction(particle(i).Position);
    kp_pso = (particle(i).Position(1));
    ki_pso = (particle(i).Position(2));
    kd_pso = (particle(i).Position(3));
    sim('simpsojoe')
    r=simout.Data;
    IAE = r(end);
    particle(i).Cost=IAE;

    %Update The Personal Best
    particle(i).Best.Position = particle(i).Position;
    particle(i).Best.Cost = particle(i).Cost;

    %Update The Global Best
    if particle(i).Best.Cost < GlobalBest.Cost
        GlobalBest = particle(i).Best;
    end
end
%Array to Hold the Best Value on Each Iteration
BestCosts = zeros(maxIt,1);

```



```

%% Main Loop of PSO
for it=1:maxIt
    for i=1:nPop
        particle(i).Velocity = w*particle(i).Velocity...
            + C1*rand(VarSize).*(particle(i).Best.Position-
particle(i).Position)...
            + C2*rand(VarSize).*(GlobalBest.Position-
particle(i).Position);
        particle(i).Position = particle(i).Position + particle(i).Velocity;
        %particle(i).Cost = CostFunction(particle(i).Position);
        if (particle(i).Position(1)) > kp_max
            particle(i).Position(1) = kp_max;
        else
            if(particle(i).Position(1)) < kp_min
                particle(i).Position(1) = kp_min;
            end
        end
        kp_pso = particle(i).Position(1);

        if (particle(i).Position(2)) > ki_max
            particle(i).Position(2) = ki_max;
        else
            if(particle(i).Position(2)) < ki_min
                particle(i).Position(2) = ki_min;
            end
        end
        ki_pso = particle(i).Position(2);

        if (particle(i).Position(3)) > kd_max
            particle(i).Position(3) = kd_max;
        else
            if(particle(i).Position(3)) < kd_min
                particle(i).Position(3) = kd_min;
            end
        end
    end
end

```

```

        end
    end
    kd_pso = particle(i).Position(3);

    sim('simpsojoe')
    r=simout.Data;
    IAE = r(end);
    particle(i).Cost=IAE;
    if particle(i).Cost < particle(i).Best.Cost
        particle(i).Best.Position = particle(i).Position;
        particle(i).Best.Cost = particle(i).Cost;

        if particle(i).Best.Cost < GlobalBest.Cost
            GlobalBest = particle(i).Best;
        end
    end

end

end
BestCost(it) = GlobalBest.Cost;
w = w * wdamp;
end

```

LAMPIRAN B Program Arduino

```
#include <Servo.h>

Servo myservo;
Servo myservo1;
Servo myservo2;
float pos = 0;
float Kp = 0.0239069;
float Ki = 0.0001382;
float Kd = 0.0535180;
float P,I,D,PID;
int sp = 40;
float error;
float errorlalu = 0;
float jumlaherror=0;
int waktu=0;// input derajat
int encoderOPinA=2;
int encoderOPos = 0;
int encoderOPinALast = LOW;
int n = LOW;
int RPM = 0;
unsigned long lastmillis=0;

void setup()
{
  myservo.attach(8);
  myservo1.attach(9);
  myservo2.attach(10);
  pinMode (encoderOPinA,INPUT);
  Serial.begin (9600);
}
void loop()
{ n = digitalRead(encoderOPinA);
```

```

    if (millis() - lastmillis == 1000)
    { RPM = encoder0Pos*3.75;
      error = sp - RPM;
      P = Kp * error;
      I = Ki * (error + jumlaherror);
      D = Kd * (error - errorlalu);
      PID = P + I + D;
      jumlaherror += error;
      errorlalu = error;
      pos = PID;
      Serial.print("Waktu=\t");
      Serial.print(waktu);
      Serial.print("\tRPM=\t");
      Serial.print("\tSudut=\t");
      Serial.println(pos);
      encoder0Pos = 0;
      lastmillis = millis();
      waktu++;
      if (waktu >= 1000)
      {sp = 40;}
      else if (waktu >= 500)
      {sp = 80;}
    }
    if ((encoder0PinALast == LOW) && (n == HIGH))
    { encoder0Pos++; }
    encoder0PinALast = n;
    myservo.write(pos);
    myservo1.write(pos);
    myservo2.write(pos);
  }

```

**LAMPIRAN C Respon *Real Plant* Turbin Angin dengan
Input Sudut Motor Servo 4⁰ dan
Kecepatan Angin 2,6 m/s**

Tabel C1 Respon *Real Plant* Turbin Angin Sumber Kecepatan
Angin 2,6 m/s

t (s)	rpm
0	8
1	8
2	11
3	11
4	8
5	15
6	11
7	11
8	11
9	15
10	11
11	15
12	11
13	11
14	15
15	15
16	11
17	11
18	15
19	11
20	15
21	15
22	11

t (s)	rpm
23	11
24	15
25	11
26	15
27	11
28	11
29	11
30	15
31	11
32	11
33	15
34	11
35	13
36	13
37	13
38	9
39	9
40	9
41	9
42	11
43	11
44	11
45	9

Tabel C1 Respon *Real Plant* Turbin Angin Sumber
Kecepatan Angin 2,6 m/s (Lanjutan)

t (s)	rpm
46	9
47	10
48	9
49	11
50	10
51	13
52	13
53	17
54	17
55	16
56	16
57	12
58	12
59	15
60	11
61	12
62	14
63	12
64	12
65	11
66	15
67	13
68	12
69	16
70	15
71	17

t (s)	rpm
72	17
73	17
74	13
75	12
76	14
77	14
78	14
79	16
80	13
81	16
82	13
83	13
84	13
85	13
86	15
87	16
88	16
89	13
90	16
91	16
92	16
93	16
94	16
95	13
96	16
97	15

Tabel C1 Respon *Real Plant* Turbin Angin Sumber
Kecepatan Angin 2,6 m/s (Lanjutan)

t (s)	rpm
98	14
99	14
100	14
101	14
102	14
103	15
104	15
105	20
106	16
107	16
108	20
109	16
110	16
111	20
112	15
113	15
114	20
115	15
116	15
117	19
118	19
119	19
120	20
121	20
122	20
123	16

t (s)	rpm
124	20
125	20
126	19
127	16
128	20
129	19
130	19
131	19
132	15
133	19
134	19
135	19
136	19
137	20
138	16
139	20
140	20
141	20
142	20
143	24
144	20
145	20
146	20
147	19
148	23
149	19

Tabel C1 Respon *Real Plant* Turbin Angin Sumber
Kecepatan Angin 2,6 m/s (Lanjutan)

t (s)	rpm
150	20
151	24
152	20
153	24
154	21
155	25
156	21
157	25
158	22
159	26
160	22
161	26
162	23
163	27
164	23
165	28
166	23
167	26
168	21
169	25
170	25
171	21
172	25
173	25
174	20
175	26

t (s)	rpm
176	27
177	23
178	25
179	25
180	25
181	21
182	25
183	27
184	23
185	26
186	27
187	31
188	31
189	30
190	31
191	27
192	28
193	30
194	30
195	32
196	32
197	31
198	30
199	30
200	29
201	29

Tabel C1 Respon *Real Plant* Turbin Angin Sumber
Kecepatan Angin 2,6 m/s (Lanjutan)

t (s)	rpm
202	29
203	33
204	33
205	29
206	33
207	31
208	31
209	31
210	30
211	30
212	31
213	33
214	31
215	33
216	31
217	33
218	33
219	31
220	34
221	39
222	39
223	39
224	37
225	37
226	39
227	41

t (s)	rpm
228	37
229	43
230	47
231	49
232	47
233	53
234	49
235	52
236	52
237	49
238	50
239	49
240	50
241	51
242	53
243	55
244	58
245	59
246	58
247	65
248	65
249	60
250	66
251	62
252	64
253	60

Tabel C1 Respon *Real Plant* Turbin Angin Sumber
Kecepatan Angin 2,6 m/s (Lanjutan)

t (s)	rpm
254	65
255	63
256	67
257	63
258	61
259	62
260	63
261	63
262	63
263	66
264	68
265	66
266	64
267	65
268	63
269	63
270	67
271	67
272	70
273	66
274	69
275	70
276	70
277	70
278	70
279	68

t (s)	rpm
280	68
281	71
282	71
283	76
284	76
285	76
286	77
287	77
288	77
289	77
290	77
291	77
292	77
293	79
294	77
295	81
296	78
297	81
298	81
299	85
300	81
301	81
302	85
303	82
304	86
305	86

Tabel C1 Respon *Real Plant* Turbin Angin Sumber
Kecepatan Angin 2,6 m/s (Lanjutan)

t (s)	rpm
306	87
307	82
308	86
309	86
310	88
311	88
312	88
313	87
314	87
315	87
316	86
317	89
318	89
319	87
320	87
321	91
322	90
323	90
324	90
325	90
326	92
327	92
328	95
329	91
330	94
331	97

t (s)	Rpm
332	94
333	97
334	97
335	101
336	101
337	100
338	100
339	100
340	100
341	104
342	100
343	104
344	104
345	100
346	103
347	98
348	98
349	102
350	102
351	102
352	102
353	105
354	105
355	105
356	105
357	109

Tabel C1 Respon *Real Plant* Turbin Angin Sumber
Kecepatan Angin 2,6 m/s (Lanjutan)

t (s)	rpm
358	109
359	105
360	109
361	109
362	109
363	104
364	104
365	108
366	101
367	104
368	104
369	104
370	104
371	104
372	101
373	105
374	105
375	105
376	109
377	105
378	109
379	105
380	109
381	109
382	109
383	109

t (s)	rpm
384	113
385	109
386	109
387	109
388	110
389	110
390	110
391	110
392	110
393	114
394	110
395	110
396	110
397	114
398	111
399	111
400	111
401	111
402	111
403	111
404	111
405	111
406	111
407	111
408	107
409	111

Tabel C1 Respon *Real Plant* Turbin Angin Sumber
Kecepatan Angin 2,6 m/s (Lanjutan)

t (s)	rpm
410	115
411	111
412	112
413	112
414	116
415	112
416	116
417	112
418	116
419	112
420	116
421	112
422	116
423	112
424	116
425	112
426	108
427	112
428	112
429	112
430	112
431	112
432	116
433	112
434	112
435	112

t (s)	rpm
436	112
437	116
438	112
439	116
440	116
441	112
442	112
443	116
444	112
445	112
446	112
447	108
448	112
449	112
450	112
451	112
452	112
453	116
454	112
455	116
456	112
457	112
458	108
459	112
460	108
461	108

Tabel C1 Respon *Real Plant* Turbin Angin Sumber
Kecepatan Angin 2,6 m/s (Lanjutan)

t (s)	Rpm
462	112
463	108
464	112
465	108
466	112
467	108
468	112
469	112
470	112
471	112
472	112
473	112
474	112
475	112
476	112
477	112
478	112
479	112
480	112
481	112
482	108
483	112
484	112
485	116
486	112
487	112

t (s)	rpm
488	112
489	112
490	112
491	112
492	116
493	112
494	112
495	112
496	112
497	116
498	112
499	116
500	112
501	112
502	116
503	112
504	112
505	112
506	112
507	112
508	108
509	108
510	112
511	105
512	108
513	108

Tabel C1 Respon *Real Plant* Turbin Angin Sumber
Kecepatan Angin 2,6 m/s (Lanjutan)

t (s)	rpm
514	108
515	108
516	108
517	105
518	108
519	108
520	108
521	108
522	108
523	108
524	108
525	108
526	108
527	112
528	112
529	108
530	112
531	112
532	112
533	112
534	112
535	112
536	112
537	116
538	112
539	116

t (s)	rpm
540	112
541	116
542	112
543	116
544	112
545	112
546	112
547	112
548	108
549	108
550	108
551	108
552	105
553	108
554	105
555	105
556	108
557	105
558	105
559	105
560	108
561	105
562	105
563	105
564	108
565	108

Tabel C1 Respon *Real Plant* Turbin Angin Sumber
Kecepatan Angin 2,6 m/s (Lanjutan)

t (s)	rpm
566	105
567	108
568	108
569	108
570	108
571	108
572	108
573	112
574	108
575	112
576	108
577	112
578	108
579	112
580	108
581	112
582	112
583	108
584	112
585	108
586	112
587	112
588	108
589	112
590	112
591	112

t (s)	rpm
592	112
593	112
594	112
595	112
596	112
597	112
598	108
599	112
600	112
601	112
602	108
603	112
604	112
605	108
606	112
607	112
608	112
609	112
610	112
611	116
612	112
613	112
614	112
615	112
616	108
617	112

Tabel C1 Respon *Real Plant* Turbin Angin Sumber
Kecepatan Angin 2,6 m/s (Lanjutan)

t (s)	rpm	t (s)	rpm
618	108	633	108
619	108	634	108
620	108	635	108
621	105	636	112
622	108	637	108
623	108	638	108
624	105	639	112
625	105	640	108
626	108	641	108
627	105	642	112
628	108	643	112
629	105	644	108
630	108	645	112
631	108	646	108
632	105	647	112

Rata-rata *steady state* = 110 rpm

Rata-rata respon saat *delay time* = 10 rpm

Sehingga,

rpm saat 63% dari respon *steady state* adalah:

$$63\% \times (110 - 10) + 10 = 73 \text{ rpm}$$

rpm saat 28% dari respon *steady state* adalah:

$$28\% \times (110 - 10) + 10 = 38 \text{ rpm}$$

BIODATA PENULIS



Jauharotul Maknunah, dilahirkan di Gresik, 9 Desember 1992. Tamat Sekolah Dasar di Madrasah Ibtidaiyah Islamiyah Ujungpangkah, Gresik pada tahun 2005. Sekolah Lanjutan Tingkat Pertama (SMP) diselesaikan pada tahun 2008 di MTs Islamiyah Ujungpangkah, Gresik. Lulus Sekolah Menengah Atas di SMAN 1 Sidayu, Gresik pada tahun 2011. Mulai kuliah pada tahun 2011, di jurusan D3 Teknik Elektronika – Politeknik Elektronika Negeri Surabaya. Melanjutkan Kuliah S1 Lintas Jalur di Jurusan Teknik Fisika, ITS mulai tahun 2015 hingga sekarang.